



Universidad  
Carlos III de Madrid

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

# Desarrollo de un asistente oral para la gestión de llamadas y mensajes en dispositivos Android

Autor: Carlos Molina López

Tutor: Dr. David Griol Barres

Leganés, septiembre de 2013



Título: Desarrollo de un asistente oral para la gestión de llamadas y mensajes en dispositivos Android

Autor: Carlos Molina López

Tutor: Dr. David Griol Barres

## EL TRIBUNAL

Presidente: \_\_\_\_\_

Vocal: \_\_\_\_\_

Secretario: \_\_\_\_\_

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día \_\_ de \_\_\_\_\_ de 20\_\_ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE



# Agradecimientos

En primer lugar le quería agradecer a mi tutor, David Griol, por darme la oportunidad de involucrarme en el mundo de los sistemas de diálogo y los sistemas operativos móviles, pero sobre todo por su dedicación y su profesionalidad en la dirección del Trabajo Fin de Grado, además de haberme guiado y animado en cada momento.

Mención especial merecen mis padres, quienes han hecho tuyas mis tristezas y alegrías. Gracias por vuestra constancia y vuestra paciencia, por vuestro interés y dedicación, y por inculcarme la importancia y el peso que tienen los estudios en la vida, pues sin esa educación seguramente no habría sido capaz de realizar una carrera.

Quiero destacar a mi hermana, Esther, a la que siempre he tomado como un ejemplo a seguir. Gracias por estar siempre ahí, por ofrecerme tu ayuda siempre que la he necesitado y por darme esa fuerza para seguir adelante en los momentos más difíciles. Por ello, parte de este triunfo también es tuyo.

Tengo muy presente a todos mis tíos y primos, por quienes me he sentido respaldado en todo momento. Gracias por vuestro apoyo y cariño y por las palabras de ánimo cuando más lo necesitaba. En especial a ti, Antonio, por ser un hermano más que un primo.

No puedo olvidarme de mis abuelas, Ascensión y Faustina, gracias por vuestro incondicional apoyo a lo largo de todos estos años. Me alegra enormemente poder compartir con vosotras la satisfacción de ver terminados mis estudios superiores.

A mis amigas y amigos, por haber estado ahí en los buenos tiempos, pero sobre todo por haberme apoyado en los momentos de bajón. Por ello, no podéis faltar en este apartado Alex, Marina, Rocío, Ángel, Ramón, etc.

Gracias a los compañeros y amigos de la universidad, con los que he compartido tantísimas horas de estudio y de prácticas. Sé que con muchos de vosotros mantendré una larga amistad.

En definitiva, muchísimas gracias a todos lo que habéis confiado en mí, pues sin vuestro apoyo no habría alcanzado esta meta.



# Resumen

El objetivo principal de este Trabajo Fin de Grado es el desarrollo de una aplicación que facilite una comunicación más natural con los teléfonos inteligentes y resto de dispositivos móviles con el sistema operativo Android. A través de la aplicación desarrollada, los usuarios pueden realizar funciones básicas en estos dispositivos móviles utilizando tanto su voz como el teclado. Del mismo modo, la aplicación utiliza tanto la voz como funcionalidades gráficas para la comunicación con los usuarios.

Entre las funcionalidades de la aplicación desarrollada se incluyen la consulta de la agenda y búsqueda de contactos tanto por su nombre como por su número telefónico, la incorporación, edición y borrado de contactos, llamar o enviar mensajes, consultar la bandeja de mensajes, cambiar la contraseña de la aplicación y determinar si se quiere que la aplicación interactúe con el usuario de forma oral o no.

Además, el programa se apoya en otras tecnologías adicionales, como es la utilización de bases de datos (SQLite), uso de diferentes lenguajes de programación (Java, SQL, XML) y servicios propios de Android (Google Voice Recognizer y Text To Speech), que la hacen más dinámica y flexible, aumentando su calidad y eficiencia.

El Trabajo Fin de Grado se complementa con un análisis detallado de los sistemas operativos móviles, centrándose especialmente en Android. También se realiza un estudio acerca de los sistemas de diálogo. Estos estudios complementarios se han realizado con el objetivo de aplicar todo su potencial para completar con éxito el Trabajo Fin de Grado.

**Palabras clave:** Sistemas operativos móviles, Android, sistemas de diálogo, dispositivos móviles, interacción oral, aplicación, bases de datos.





# Abstract

The main objective of this Bachelor Project is the development of an application that can make it easier to have a more natural communication with the smart phones and all mobile devices that have the Android operative system. Through the developed application, users can make basic actions in those devices by using both their voices and keyboards. In the same way, the application makes use of the voice and graphic functions to interact with the users.

Among all the functionality of this application it is included to check the agenda and search contacts by name or telephone number, add, edit or delete contacts, call and send messages, check the message inbox, change our personal password for the application, or even choose whether or not using an oral interaction with it.

Moreover, the program is based on additional technologies, as can be the use of data bases (SQLite), different programming languages (Java, SQL, XML) and Android services (Google Voice Recognizer, Text to Speech), making the development more dynamic and flexible, increasing his quality and efficiency.

The Bachelor Project is complemented with a detailed analysis of the mobile operative systems, focusing specially on Android. Additionally, it is included a research about dialogue systems. Those supplementary investigations have been made with the intention of taking advantage of all their potential so as to conclude the Bachelor Project successfully.

**Key words:** mobile operative systems, Android, dialogue systems, mobile devices, oral interaction, application, data bases.



# Índice general

<b>1. Introducción .....</b>	<b>1</b>
1.1 Antecedentes.....	1
1.2 Objetivos .....	4
1.3 Fases de desarrollo .....	5
1.4 Planificación temporal.....	7
1.5 Medios y documentación utilizados .....	9
1.6 Estructura de la memoria.....	10
<b>2. Estado del arte .....</b>	<b>13</b>
2.1 Sistemas operativos móviles.....	13
2.1.1 Sistemas operativos móviles más destacados.....	14
2.1.2 Elección del sistema operativo móvil.....	19
2.2 Android.....	21
2.2.1 Arquitectura de Android.....	21
2.2.2 Versiones de Android y niveles de API .....	24
2.2.3 Android 2.3 vs. Android 4.0.....	29
2.3 Los sistemas de diálogo.....	30
2.3.1 Arquitectura de un sistema de diálogo .....	30
<b>3. Descripción general del sistema .....</b>	<b>33</b>
3.1 Presentación del sistema.....	33
3.2 Herramientas del sistema.....	35
3.2.1 Eclipse .....	35
3.2.1.1 SDK Android .....	35
3.2.1.2 Sistema de ficheros.....	36
3.2.2 SQLite .....	37
3.3 Recursos necesarios para el sistema .....	38
3.3.1 Google Voice Recognizer.....	38
3.3.2 Text To Speech.....	38
3.3.3 Broadcast Receiver.....	39

3.4 Implementación de las operaciones generales .....	40
3.4.1 Gramáticas .....	40
3.4.2 Gestión de Base de Datos .....	40
3.4.2.1 Cifrado y descifrado de la base de datos .....	43
<b>4. Descripción detallada de los módulos del sistema .....</b>	<b>45</b>
4.1 Módulo Inicio .....	45
4.1.1 Funcionalidad .....	45
4.1.2 Arquitectura.....	46
4.1.2.1 Registro .....	46
4.1.2.2 Validación .....	47
4.1.3 Escenarios de uso .....	48
4.2 Módulo Contactos .....	49
4.2.1 Funcionalidad .....	49
4.2.2 Arquitectura.....	50
4.2.3 Escenarios de uso .....	51
4.3 Módulo Bandeja Mensajes .....	52
4.3.1 Funcionalidad .....	52
4.3.2 Arquitectura.....	54
4.3.2.1 Enviar mensaje a un contacto existente de la bandeja.....	54
4.3.2.2 Enviar mensaje introduciendo número telefónico.....	55
4.3.3 Escenarios de uso .....	56
4.4 Módulo Frecuentes .....	57
4.4.1 Funcionalidad .....	57
4.4.2 Arquitectura.....	58
4.4.3 Escenarios de uso .....	59
4.5 Módulo Configuración .....	59
4.5.1 Funcionalidad .....	59
4.5.2 Arquitectura.....	61
4.5.2.1 Habilidad de la opción de audio .....	61
4.5.2.2 Cambiar contraseña.....	62
4.5.3 Escenarios de uso .....	63
4.6 Módulo Llamar .....	64
4.6.1 Funcionalidad .....	64
4.6.2 Arquitectura.....	64
4.6.3 Escenarios de uso .....	65

4.7 Módulo Añadir Contacto .....	66
4.7.1 Funcionalidad .....	66
4.7.2 Arquitectura.....	67
4.7.3 Escenarios de uso .....	68
4.8 Módulo Editar Contacto .....	69
4.8.1 Funcionalidad .....	69
4.8.2 Arquitectura.....	70
4.8.3 Escenarios de uso .....	71
4.9 Módulo Borrar Contacto.....	72
4.9.1 Funcionalidad .....	72
4.9.2 Arquitectura.....	72
4.9.3 Escenarios de uso .....	73
4.10 Módulo Enviar Mensaje .....	74
4.10.1 Funcionalidad .....	74
4.10.2 Arquitectura.....	75
4.10.3 Escenarios de uso .....	77
4.11 Módulo Recibir Mensaje .....	77
4.11.1 Funcionalidad .....	77
<b>5. Evaluación de la aplicación.....</b>	<b>79</b>
5.1 Metodología de evaluación.....	79
5.2 Resultados de la evaluación.....	82
<b>6. Conclusiones y trabajo futuro.....</b>	<b>89</b>
6.1 Conclusiones .....	89
6.2 Trabajo futuro.....	93
<b>Presupuesto .....</b>	<b>95</b>
<b>Glosario.....</b>	<b>99</b>
<b>Manual de instalación .....</b>	<b>103</b>
<b>Bibliografía .....</b>	<b>107</b>



# Índice de figuras

Figura 1.1 Diagrama WBS de las tareas definidas del Trabajo Fin de Grado.....	6
Figura 1.2 Diagrama de Gantt de la planificación temporal del Trabajo Fin de Grado .....	8
Figura 2.1 Comparativa plataformas móviles .....	19
Figura 2.2 Cuota de mercado de los sistemas operativos para móviles .....	20
Figura 2.3 Arquitectura de Android .....	21
Figura 2.4 Utilización de dispositivos Android según plataforma instalada.....	29
Figura 2.5 Arquitectura modular de un sistema de diálogo .....	31
Figura 3.1 Servicios ofrecidos por la aplicación .....	34
Figura 3.2 Sistema de ficheros de la aplicación desarrollada.....	36
Figura 3.3 Inicialización y uso del conversor de texto a voz .....	39
Figura 3.4 Base de datos de la aplicación desarrollada.....	40
Figura 4.1 Interfaz correspondiente al módulo Inicio .....	46
Figura 4.2 Escenarios de uso del módulo Inicio. Registro de usuario .....	48
Figura 4.3 Escenarios de uso del módulo Inicio. Validación de usuario .....	48
Figura 4.4 Interfaz correspondiente al módulo Contactos .....	49
Figura 4.5 Escenarios de uso del módulo Contactos. Editar Contacto.....	51
Figura 4.6 Escenarios de uso del módulo Contactos. Añadir Contacto .....	52
Figura 4.7 Interfaz correspondiente al módulo Bandeja Mensajes .....	53
Figura 4.8 Escenarios de uso del módulo Bandeja Mensaje. Enviar mensaje a un contacto existente.....	56
Figura 4.9 Escenarios de uso del módulo Bandeja Mensaje. Enviar mensaje introduciendo el número telefónico.....	56
Figura 4.10 Interfaz correspondiente al módulo Frecuentes .....	57
Figura 4.11 Escenario de uso del módulo Frecuentes. Llamar a un contacto .....	59
Figura 4.12 Interfaz principal del módulo Configuración.....	60
Figura 4.13 Interfaz correspondiente al cambio de contraseña del módulo Configuración .....	61
Figura 4.14 Escenarios de uso del módulo Configuración. Habilitar audio.....	63
Figura 4.15 Escenarios de uso del módulo Configuración. Cambiar contraseña .....	64
Figura 4.16 Escenario de uso del módulo Llamar.....	65
Figura 4.17 Interfaz correspondiente al módulo Añadir Contacto .....	66
Figura 4.18 Escenario de uso del módulo Añadir Contacto.....	68
Figura 4.19 Interfaz correspondiente al módulo Editar Contacto .....	69
Figura 4.20 Escenario de uso del módulo Editar Contacto .....	71

Figura 4.21 Escenario de uso del módulo Borrar Contacto .....	73
Figura 4.22 Interfaz correspondiente al módulo Enviar Mensaje .....	74
Figura 4.23 Escenario de uso del módulo Enviar Mensaje .....	77
Figura 4.24 Notificación generada al recibir un mensaje de texto .....	78
Figura 5.1 Cuestionario desarrollado para la evaluación objetiva de la aplicación .....	82
Figura 5.2 Respuesta de los participantes a la pregunta 1 .....	83
Figura 5.3 Respuesta de los participantes a la pregunta 2 .....	83
Figura 5.4 Respuesta de los participantes a la pregunta 3 .....	84
Figura 5.5 Respuesta de los participantes a la pregunta 4 .....	84
Figura 5.6 Respuesta de los participantes a la pregunta 5 .....	85
Figura 5.7 Respuesta de los participantes a la pregunta 6 .....	85
Figura 5.8 Respuesta de los participantes a la pregunta 7 .....	86
Figura 5.9 Respuesta de los participantes a la pregunta 8 .....	86
Figura 5.10 Respuesta de los participantes a la pregunta 9 .....	87
Figura 5.11 Respuesta de los participantes a la pregunta 10 .....	87
Figura 7.1 Detalle de costes de recursos humanos del Trabajo Fin de Grado .....	97
Figura 7.2 Detalle del coste total del Trabajo Fin de Grado .....	97
Figura 9.1 Interfaz de un dispositivo Android con acceso a Google Play Store .....	103
Figura 9.2 Programas ofrecidos por Google Play Store .....	104
Figura 9.3 Búsqueda en Google Play Store por categoría .....	104
Figura 9.4 Instalación de la aplicación desarrollada .....	105
Figura 9.5 Permisos de la aplicación desarrollada .....	105
Figura 9.6 Acceso directo a la aplicación desarrollada .....	106



# Capítulo 1

## Introducción

En este primer capítulo se presentan los antecedentes del Trabajo Fin de Grado, se establecen los objetivos que se definieron para el mismo y se describen los aspectos generales relativos a las fases de desarrollo, la planificación temporal y los medios empleados para su consecución. En último lugar, se detalla la estructura de los capítulos que conforman este documento.

### 1.1 Antecedentes

En la actualidad, los dispositivos móviles y los ordenadores se han convertido en una parte indispensable de nuestras vidas. Gracias a ellos, tenemos la posibilidad de acceder a información y a funcionalidades que hace no muchos años eran totalmente impensables. Los usuarios desean acceder a estos servicios de forma fácil y eficiente en cualquier lugar y momento, por lo que se busca el desarrollo de interfaces que establezcan medios de comunicación entre dichos usuarios y las máquinas.

Debido a que la voz es uno de los modos más intuitivos y naturales para interactuar y comunicarse entre los seres humanos, las aplicaciones basadas en sistemas de diálogo [MCT04] [PIE12] se han convertido en una de las principales opciones para dotar a los dispositivos de capacidad de comunicación.

El objetivo de estos programas informáticos es el de proporcionar un determinado servicio a los usuarios, interactuando con ellos de forma oral o multimodal. Algunos ejemplos de servicios que se ofrecen de forma oral son las centralitas telefónicas, información y reserva de viajes o el control de dispositivos domóticos.

En los últimos años, los servicios de voz están sufriendo una gran evolución y expansión, y su implantación en servicios diarios es cada día más frecuente. Esta evolución de los servicios de voz se debe en gran medida a la explosión de los dispositivos móviles, y sobre todo a las numerosas ventajas que ofrecen: facilidad a la hora de realizar tareas de forma automática ya que son más intuitivos, efectividad y mayor rapidez.

Según los resultados del estudio publicado por Online Business School sobre “El uso de las tecnologías de la información y la comunicación en los hogares – 2013” [OBS13], el 95,9% de los hogares españoles poseen al menos un teléfono móvil. De ese porcentaje el 59,3% teléfonos inteligentes (*smartphone*), cifra que va en aumento ya que el 81% de los teléfonos móviles adquiridos en España en 2012 fueron teléfonos inteligentes. Además, de todos los usuarios que poseen un *smartphone*, el 23,5% también tiene una tablet-PC. Estos datos convierten a España en el país con mayor penetración de estos dispositivos móviles inteligentes entre Alemania, Francia, Italia y Reino Unido. Por último, del estudio también se concluye que el 25,2% de las acciones que se realizan con este tipo de dispositivos están relacionadas con el envío de mensajería y chats, convirtiéndose en la principal acción que se realiza con dispositivos móviles inteligentes.

Estos datos ponen de manifiesto que cada vez más usuarios se decantan por la utilización del teléfono y el acceso por voz antes que por otros medios, ya que en el último año se adquirieron casi 4 millones de dispositivos móviles inteligentes, pudiéndolos utilizar para la consulta oral de la información mediante los sistemas de diálogo.

Entre las aplicaciones ya existentes que ofrecen estos servicios, la gran mayoría permiten únicamente realizar una serie de acciones por medio de la voz, sin existir la reciprocidad de que la aplicación se comunique también de forma oral con los usuarios para acciones tan simples como comunicarles con qué persona se están enviando un mensaje, o cual es el contenido del mensaje. A continuación se enumeran una serie de aplicaciones que hacen uso del servicio de reconocimiento de voz para realizar acciones demandadas por el usuario:

**Siri:** Es la principal aplicación dentro de este ámbito. Responde preguntas del usuario y hace recomendaciones, entre otras funcionalidades, utilizando procesamiento del lenguaje natural. Sin embargo, esta aplicación está disponible únicamente para dispositivos móviles inteligentes cuyo sistema operativo es iOS, por lo que el rango de usuarios que puedan adquirir esta aplicación es bastante reducido [SIRI].

Teniendo en cuenta únicamente aplicaciones desarrolladas, como mínimo, para dispositivos móviles que posean el sistema operativo móvil de Android [ANDROIDLIBRE], las principales son:

**Cloe**: Es la alternativa en Android que más se puede parecer a la anteriormente mencionada Siri. Se trata de un asistente virtual por voz en español que cuenta con decenas de recursos con información preparada de antemano, como cambio de divisas, horarios y tarifas de Renfe, programación de la televisión nacional, etc. De esta forma, esta aplicación puede responder a preguntas como el horario de un tren determinado o la previsión meteorológica de una ciudad española. Sin embargo, no se trata de una aplicación gratuita, sino que el usuario debe pagar 1,49€ para poder instalarla en su dispositivo.

**Jeannie**: Se trata de una aplicación desarrollada por Pannous, la misma empresa que lanzó la aplicación Voice Actions. Este programa soporta los principales idiomas europeos: español, alemán, inglés, etc. Entre sus principales funcionalidades cabe destacar que permite realizar búsquedas por Internet, fijar alarmas, escuchar música determinada y acceder a traducciones. Existe una versión gratuita y otra de pago, que elimina la publicidad y desbloquea una serie de funciones con respecto a la gratuita.

**Edwin**: Podría catalogarse como una alternativa a la anteriormente mencionada Jeannie. Permite consultar el tiempo, la hora, llamar y enviar mensajes. Asimismo da la posibilidad de realizar búsquedas por internet. El principal inconveniente de esta aplicación es que únicamente soporta el inglés como idioma para comunicarse.

También hay que mencionar otras aplicaciones que se encuentran en su versión beta (fase de pruebas) como puedan ser Vita, y Sherpa. Las acciones que pueden realizar estas dos aplicaciones son muy similares, ya que permiten al usuario enviar mensajes cortos y e-mails por voz, así como actualizar el estado de ciertas redes sociales.

Mediante la aplicación a desarrollar en este Trabajo Fin de Grado, se llevarán a cabo las principales acciones que se han mencionado que realizan las aplicaciones de reconocimiento de voz. Además, se creará una mayor interactividad entre el asistente de voz y el usuario, convirtiendo la comunicación en un proceso mucho más humanizado.

También cabe destacar que, mediante la aplicación desarrollada el usuario podrá realizar los servicios que se le ofrecen no sólo mediante voz, sino también con el teclado el dispositivo, pudiendo combinar ambas opciones, lo que le otorga un mayor dinamismo a la aplicación.

Por último, a diferencia de las mencionadas, la información resultante guardará cierta información del usuario para poder aconsejarle o mostrarle sobre ciertos aspectos, como por ejemplo aquellos contactos con los que el usuario se comunica con mayor frecuencia.

## 1.2 Objetivos

Como se ha mencionado en el apartado anterior, el objetivo principal de este Trabajo Fin de Grado es el desarrollar una aplicación en Android con la que los usuarios finales puedan interactuar tanto de forma oral como mediante el teclado de sus dispositivos móviles.

Aprovechando las inmensas opciones que ofrece el sistema operativo Android, se ha implementado una aplicación útil, que puede ser utilizada por los usuarios para resolver sus necesidades diarias de forma eficaz.

De esta forma, los usuarios de la aplicación podrán realizar de forma oral las siguientes acciones: llamar o enviar mensajes de texto, consultar su agenda así como su bandeja de mensajes, añadir, editar o eliminar cualquier contacto de su agenda telefónica, etc. También tendrán la posibilidad de poder escuchar los mensajes de texto que otros contactos les hayan enviado, así como los suyos propios.

En base al objetivo principal citado, se han definido una serie de objetivos parciales:

- Completar un análisis detallado de los principales sistemas operativos existentes para dispositivos móviles. Para ello se ha estudiado la arquitectura, el lenguaje de programación, el coste de publicación de aplicaciones, etc., de cada uno de ellos. Por último, se ha realizado un análisis de la evolución de sus respectivas cuotas de mercado en los últimos tres años.
- Llevar a cabo un estudio completo del sistema operativo elegido, que en este caso es Android, partiendo de su arquitectura hasta la cuota de mercado de cada una de las distintas versiones existentes, pasando por un análisis exhaustivo de las mejoras que incorpora cada una de dichas versiones.
- Realizar un estudio detallado de los sistemas de diálogo, centrándonos especialmente en la arquitectura y en la explicación de cada uno de los módulos de los que se compone.
- Facilitar el uso de esta aplicación a usuarios que no son expertos en el manejo de dispositivos móviles, por ejemplo, personas ancianas que no están acostumbradas a los smartphones o tablets.

- Facilitar el acceso a las funcionalidades ofrecidas en esta aplicación a personas con discapacidades visuales, contribuyendo a acercarles el mundo tecnológico a estos usuarios.

## 1.3 Fases de desarrollo

La realización de este Trabajo Fin de Grado se ha estructurado en tres fases de desarrollo bien diferenciadas que se comentan a continuación:

### **Fase 1: Planificación.**

- **Estudio de los sistemas operativos móviles.** Aproximación a los sistemas operativos móviles y determinación del sistema operativo para el que se desarrollará la aplicación a implementar.
- **Estudio del sistema operativo Android.** Aproximación al sistema operativo Android y determinación de la versión en la que se desarrollará la aplicación.
- **Estudio de los sistemas de diálogo.** Aproximación a los sistemas de diálogo.
- **Planificación y análisis de requisitos de la aplicación.** Determinación de los requisitos que cumplirá la aplicación desarrollada.
- **Estudio de las tecnologías necesarias.** Estudio de Android y SQLite.

### **Fase 2: Desarrollo.**

- **Análisis y diseño inicial.** División de las distintas funcionalidades de la aplicación en diferentes módulos y submódulos.
- **Implementación del sistema.** Desarrollo de todos los módulos y submódulos, así como de las correspondientes interfaces para facilitar la interacción entre ellos.
- **Pruebas unitarias.** Estudio, configuración y realización de pruebas funcionales para cada módulo.
- **Pruebas de integración.** Proceso de pruebas del sistema completo hasta alcanzar una versión completamente estable del mismo.
- **Evaluación de la aplicación.** Estudio de los sistemas de evaluación para interfaces orales, realización de las preguntas de la encuesta de evaluación, recogida y análisis de los resultados.

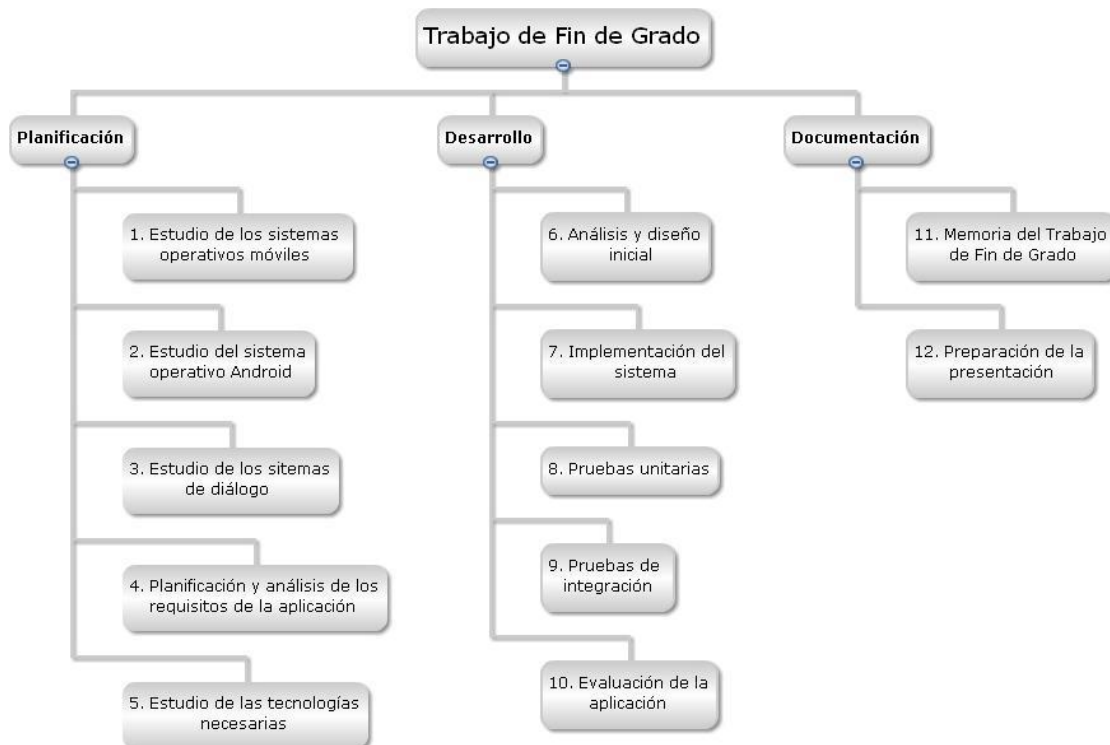
### **Fase 3: Documentación.**

- **Memoria del Trabajo de Fin de Grado.** Redacción del presente documento de memoria del Trabajo de Fin de Grado.
- **Preparación de la presentación.**

Para una mejor comprensión de las fases de desarrollo y de las tareas realizadas para cada fase, se ha construido un diagrama WBS (*Work Breakdown Structure*). Se trata de un diagrama con estructura de árbol en el que se ordenan las tareas y subtareas según un criterio determinado. La principal utilidad de este diagrama es la de proporcionar un método sistemático para confeccionar la lista de tareas.

La Figura 1.1 representa el diagrama WBS mencionado anteriormente. Como se puede observar, para la realización del Trabajo de Fin de Grado se han definido tres familias de tareas, que constituyen el primer nivel de la jerarquía a partir del cual se han desarrollado el resto, hasta alcanzar un total de 12 tareas que componen el Trabajo de Fin de Grado.

Todas las tareas pertenecientes a una misma familia (a la misma tarea del primer nivel de jerarquía), aparecen formando una columna debajo del nombre de dicha familia.



**Figura 1.1 Diagrama WBS de las tareas definidas del Trabajo Fin de Grado**

## 1.4 Planificación temporal

Una vez establecidas las fases y tareas en las que se ha dividido el Trabajo Fin de Grado, se ha generado un diagrama que vincula dichas tareas a un calendario, posibilitando de esta manera que se pueda realizar un seguimiento detallado de la evolución de cada una de las tareas citadas.

Para la planificación temporal de las fases se ha utilizado como herramienta de ayuda un diagrama de Gantt. Para la realización de dicho diagrama, además de incluir las 12 tareas que se citan en el apartado anterior y que se muestran en la Figura 1.1, se han añadido un hito de inicio de Trabajo Fin de Grado y otro hito de fin de Trabajo Fin de Grado. Estos hitos son tareas de duración nula, cuya misión es la de marcar puntos de control de avance en el Trabajo Fin de Grado.

A continuación se introducen las duraciones de cada tarea, considerando todos los días de la semana como días laborables, y estableciendo una jornada de trabajo de 6 horas. Así mismo, se han introducido las relaciones de precedencia de cada tarea, que sirven para fijar la restricción de no empezar una tarea determinada hasta que su tarea precedente no haya finalizado.

En la Figura 1.2 se puede observar el diagrama de Gantt resultante de aplicar el procedimiento de planificación explicado. Con este diagrama se muestra una visión global de la planificación del proyecto, así como de sus distintas fases y de su duración.

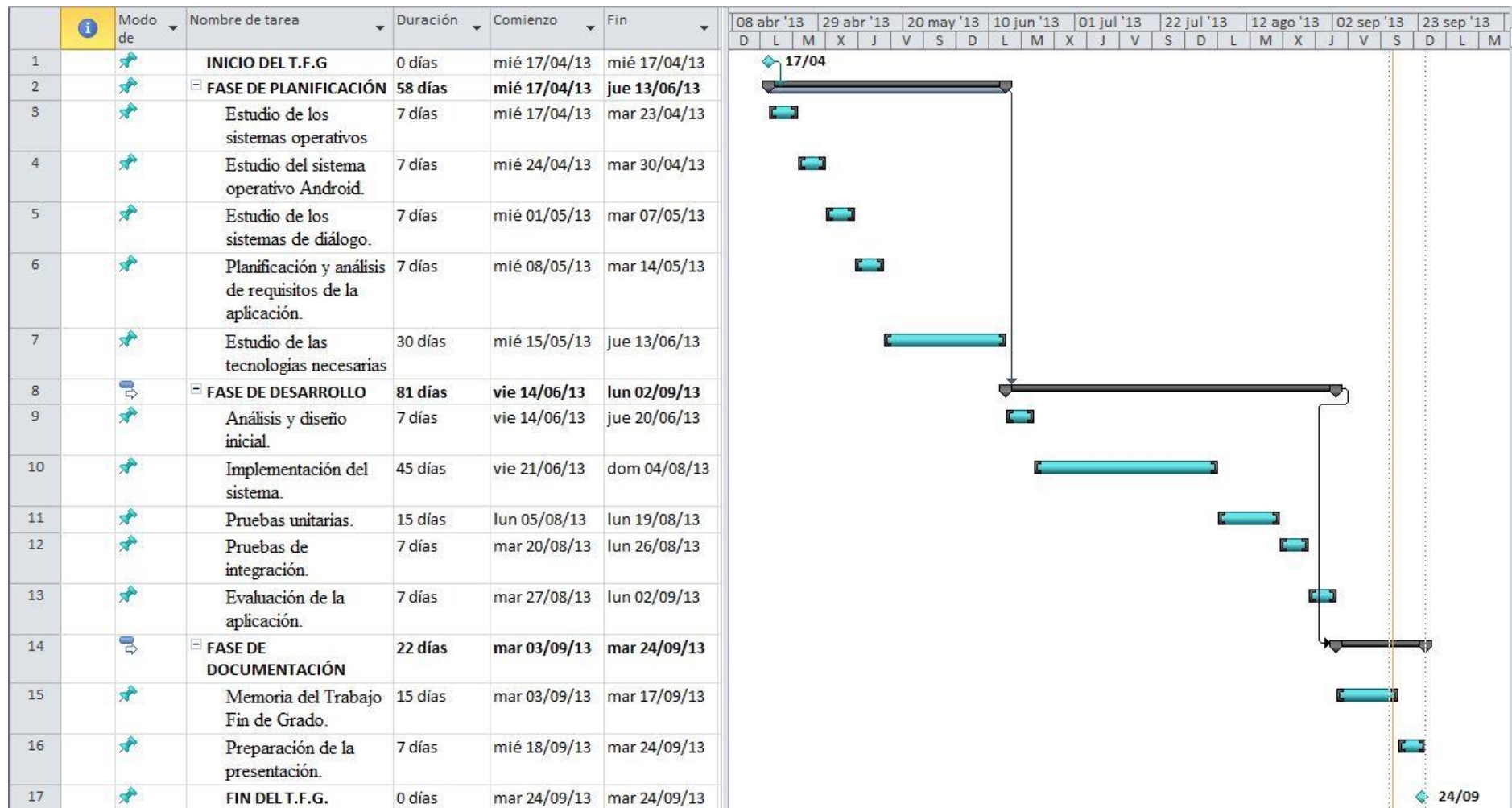


Figura 1.2 Diagrama de Gantt de la planificación temporal del Trabajo Fin de Grado



## 1.5 Medios y documentación utilizados

A continuación se enumeran los medios empleados para la realización del Trabajo de Fin de Grado.

- Dispositivos Hardware:
  - Ordenador de mesa.
  - Monitor.
  - Periféricos habituales (teclado, ratón).
  - Smartphone con Android 2.3 o superior.
  - Cable USB.
- Dispositivos Software:
  - Plataforma Eclipse con plugins de Android.
  - Paquete Microsoft Office 2010.
  - Adobe Photoshop CS6.
  - Adobe Reader.
  - Editor de programación Notepad ++.

En el apartado Presupuesto de esta memoria se facilita información relativa al coste de los medios que se acaban de citar.

En lo referente a la documentación examinada, se han empleado numerosos libros y artículos acerca del lenguaje de programación Android, así como sobre los sistemas de diálogo.

## 1.6 Estructura de la memoria

Con el objetivo de facilitar la lectura de la memoria, a continuación se incluye un breve resumen de cada uno de los capítulos que la integran.

**Capítulo 1: Introducción.** En este primer capítulo se establecen el propósito y los objetivos de todo el contenido del Trabajo Fin de Grado. Además, se incluyen las fases de desarrollo, la planificación temporal, los medios empleados y la estructura de la memoria.

**Capítulo 2: Estado del arte.** En el segundo capítulo se lleva a cabo un estudio completo de los principales sistemas operativos móviles existentes en la actualidad. Como consecuencia de dicho estudio, se elige el sistema operativo Android del cual se realiza un análisis aún más exhaustivo de sus distintas versiones con el fin de determinar en cual se desarrollará la aplicación. Por último se hace un estudio de los sistemas de diálogo.

**Capítulo 3: Descripción general del sistema.** En este capítulo se proporciona una visión global de la aplicación desarrollada, analizando las tecnologías empleadas y detallando las operaciones más comunes tratadas a lo largo de su implementación.

**Capítulo 4: Descripción detallada de los módulos del sistema.** En este capítulo se describe detalladamente cada uno de los módulos de los que consta la aplicación. Para que este capítulo sea más intuitivo, cada módulo estará formado por una interfaz distinta, teniendo los siguientes módulos: Inicial, Contactos, Añadir Contacto, Editar Contacto, Borrar Contacto, Bandeja Mensajes, Mensaje, Frecuentes y Configuración. Además, para cada uno de los módulos citados anteriormente, se realizarán una serie de pruebas unitarias para corroborar su correcto funcionamiento.

**Capítulo 5: Evaluación de la aplicación.** En este capítulo se lleva a cabo la evaluación de la aplicación a través de cuestionarios entregados a usuarios de la aplicación. A partir de los resultados obtenidos se generará una serie de gráficas para determinar la calidad de la aplicación.

**Capítulo 6: Conclusiones y trabajo futuro.** Se exponen las principales ideas, cuestiones, y conclusiones derivadas de la realización del Trabajo Fin de Grado, así como las posibles líneas de investigación que se podrían realizar a partir de la aplicación resultante.

**Presupuesto.** Este apartado contiene un análisis de los costes del diseño y desarrollo del Trabajo Fin de Grado, detallando el coste de personal y del material necesario para llevar a cabo su realización.

**Glosario.** En el apartado de Glosario se recopilan los principales términos y conceptos técnicos utilizados en la memoria, con el objetivo de facilitar al lector la comprensión de la misma.

**Manual de instalación.** En este capítulo se expondrá el manual de instalación. En dicho manual se mostrará cómo descargarse la aplicación de Google Play Store y los pasos a realizar para su correcta instalación en el dispositivo móvil.

**Bibliografía.** En este apartado se reflejan las citas bibliográficas que se han consultado para la realización tanto de la aplicación como de la memoria



# Capítulo 2

## Estado del arte

En este capítulo se analiza el contexto en el que se enmarca el Trabajo de Fin de Grado. En primer lugar, se presentan los sistemas operativos móviles, explicando las principales características de cada uno de ellos y examinando dichas características, así como su cuota de mercado, para determinar cuál se adecua mejor a los objetivos. A continuación se hace un estudio completo de las distintas versiones del sistema operativo elegido, comparándolas unas con otras y decantándonos finalmente por la que se ha considerado mejor versión para la aplicación a desarrollar. Por último, se describen de forma detallada los sistemas de diálogo, así como los módulos que los componen.

### 2.1 Sistemas operativos móviles

En los últimos años se han acentuado los avances tecnológicos, sucediéndose uno tras otro con poco espacio de tiempo entre ellos. Uno de los ejemplos más claros es la telefonía móvil. Desde los primeros años de la década pasada, los teléfonos móviles se han vuelto cada vez más y más populares y, gracias a las innovaciones tecnológicas, se han convertido en pequeños ordenadores de bolsillo, facilitándole al usuario la posibilidad de realizar distintos tipos de tareas sin necesidad de otros dispositivos.

Hoy en día los dispositivos móviles funcionan como centros multimedia, localizadores de satélites, equipos de diseño y, también, como teléfonos. Pero estos aparatos pueden llevar adelante todo esto gracias a que poseen un sistema operativo (SO) que les da la posibilidad de funcionar y realizar tareas tan distintas y complejas.

Un SO es un conjunto de programas destinados a permitir la comunicación del usuario con el hardware y a gestionar sus recursos de manera eficiente y cómoda [TAN98].

En el caso de los sistemas operativos móviles, son sistemas operativos específicos para este tipo de dispositivos que tienen en cuenta las limitaciones que se comentan a continuación:

- Limitada capacidad computacional comparada con los procesadores de los ordenadores de escritorio debido a la necesidad de ahorrar energía.
- Limitada vida de la batería debido al uso intensivo de cálculos, gráficos, sonido, conexiones inalámbricas etc.
- Limitada memoria disponible en comparación con la de un ordenador ordinario.
- Limitada capacidad de almacenamiento de datos y de visualización causada por el tamaño de las pantallas y la reducida gama de colores.

Los sistemas operativos móviles hacen posible explotar dichas características y ofrecen al programador una visión de alto nivel para realizar aplicaciones. Estas aplicaciones interactúan con el hardware gracias al sistema operativo subyacente que se constituye, por lo tanto, en un software fundamental que gobierna y controla todos los componentes del sistema.

### **2.1.1 Sistemas operativos móviles más destacados**

Hasta hace poco, a la hora de elegir un nuevo dispositivo móvil, el usuario se fijaba principalmente en su aspecto así como en la duración de su batería. Sin embargo, hoy en día el sistema operativo con el que trabaja dicho terminal es un aspecto mucho más determinante e influyente, y se ha convertido en una característica imprescindible que no se puede pasar por alto al escoger un smartphone.

La gran variedad de sistemas operativos existentes en la actualidad no se debe a una estrategia por parte de las compañías para afianzar su posición en el mercado, sino que obedece a las diferentes características, pensadas para públicos distintos, que ofrece cada sistema operativo.

A continuación se enumeran los principales sistemas operativos para dispositivos móviles, haciendo una breve explicación de cada uno de ellos.

## **Apple iOS**

Es un sistema operativo móvil desarrollado por la empresa Apple Inc. cuyo lanzamiento tuvo lugar en junio de 2007.

Se trata de una versión reducida del sistema Mac OS X para PC, que fue diseñado originalmente para iPhone, aunque después se usó en otros dispositivos móviles como iPod touch e iPad.

Este sistema operativo ha sufrido diversas evoluciones en lo que se refiere a la actualización de sus versiones. Desde la primera versión llamada iPhone OS 1.0 que llevaba instalada el primer iPhone hasta el actual iOS 7, lanzado en junio de 2013.

iOS tiene 4 capas de abstracción: la capa del núcleo del sistema operativo, la capa de servicios principales, la capa de medios de comunicación y la capa de Cocoa Touch.

El sistema operativo ocupa bastante menos de medio gigabyte del total del dispositivo, que puede ser de 8 o 16 GB. Esto se realizó para poder soportar futuras aplicaciones de Apple, así como aplicaciones de terceros publicadas en la iTunes Store o la App Store.

Para poder desarrollar aplicaciones iOS y publicarlas, hay que pagar 99\$ al año. Los desarrolladores iOS firman sus aplicaciones con un certificado emitido por Apple llamado Root CA (Root Certificate Authority), que es un certificado de autoridad del desarrollador. Este certificado sirve para garantizar que el autor es identificable y autorizado por Apple y que la aplicación no es modificada después de su creación, ya que se firma digitalmente.

El lugar del que se descargan las aplicaciones iOS es la App Store [APPSTORE]. Se trata de una de las tiendas de aplicaciones más completa y de mayor calidad, lo que supone una de las claves del éxito de los despóticos que llevan este sistema operativo.

El sistema de Apple es cerrado, por lo que hay menos posibilidades de cambiar la forma de funcionar del teléfono, así como un control más rígido de las aplicaciones publicadas. Además, únicamente se puede hacer uso de este sistema operativo con los dispositivos móviles de Apple anteriormente citados, iPod touch, iPad e iPhone, y debido a que sólo hay un fabricante y modelo, el precio de dichos dispositivos es bastante elevado.

Apple proporciona actualizaciones de iOS a través de iTunes [ITUNES]; tienda de aplicaciones multimedia desarrollada por esta misma empresa con el fin de sincronizar todos sus dispositivos móviles. Todos los parches de seguridad, mejoras o nuevas características se distribuyen mediante este programa.

## **Android**

A pesar que desde el año 2005 pertenece a Google, Android [CELULARIS] fue diseñado inicialmente por la empresa Android Inc. Este sistema operativo está basado en Linux y diseñado principalmente para dispositivos móviles con pantalla táctil.

La presentación oficial de Android tuvo lugar en el año 2007, junto a la fundación del Open Handset Alliance: alianza comercial de 84 compañías que se dedica a desarrollar estándares abiertos para dispositivos móviles.

Desde el primer móvil con Android que se vendió en octubre de 2008 hasta la actualidad, este sistema operativo ha sufrido numerosas actualizaciones. Cada nueva plataforma que se lanza es compatible con las versiones anteriores; es decir, solo se añaden nuevas funcionalidades [GIR12].

A partir de la tercera actualización, para los nombres de las sucesivas versiones se han elegido postres en orden alfabético: Cupcake, Donut, Éclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich y Jelly Bean; siendo esta última la actualización más reciente (versión 4.3), que fue anunciada en mayo 2013 y lanzada en el proyecto Android Open Source un mes después [GIR13].

Los desarrolladores Android que quieran publicar sus aplicaciones en la tienda oficial de Android, Google Play Store, tienen que pagar 25\$. Sin embargo, a diferencia de lo que ocurre con iOS, los desarrolladores de Android no necesitan un certificado emitido por Google con el que firmar las aplicaciones para garantizar que el autor es autorizado y reconocible por Google, sino que las aplicaciones son auto firmadas.

Es más difícil actualizar los dispositivos Android que los que funcionan con el sistema operativo de Apple, debido a la gran diversidad de fabricantes que lo utilizan. Por esta razón, comercialmente es más rentable dejar los móviles obsoletos.

## **Windows Phone**

Se trata de un sistema operativo móvil desarrollado por Microsoft, como sucesor de la plataforma Windows Mobile. Su lanzamiento tuvo lugar en octubre de 2010, y su última versión, llamada Windows Phone 8, fue lanzada al mercado dos años más tarde, el 29 de octubre de 2012.

Se basa en el núcleo del sistema operativo Windows CE y cuenta con un conjunto de aplicaciones básicas utilizando las APIs de Microsoft Windows. Está diseñado para ser similar estéticamente a las versiones de escritorio de Windows.

Existe una gran demanda de software de terceros disponible para Windows Phone, que se puede adquirir a través de Windows Marketplace [WINDOWSMARKET]. El mercado es accesible desde del Hub del Marketplace de los dispositivos. Está gestionado por Microsoft, que incluye un proceso de aprobación para cada aplicación de terceros, donde se evalúan varios factores estrictos de contenido y performance.



Desde marzo de 2012, Windows Marketplace está disponible en 54 países. En febrero de 2013 este mercado ya contaba con más de 50.000 apps, cifra que se ve incrementada diariamente.

Para poder desarrollar aplicaciones para esta plataforma es necesario registrarse en el App Hub. Este proceso de suscripción cuesta 99\$ y permite la publicación de aplicaciones tanto para dispositivos móviles como para la videoconsola XBOX 360.

Una vez realizado el pago, el desarrollador puede publicar un número ilimitado de aplicaciones de pago en el Marketplace. Sin embargo, tiene un máximo de publicación de cinco aplicaciones gratuitas. A partir de esta cifra, para poder publicar otra aplicación gratuita el desarrollador tendrá que hacer frente a un coste de 19,99\$ por cada aplicación gratuita extra que desee publicar.

A pesar de que marcas de como Samsung, LG o HTC incorporan en algunos de sus dispositivos móviles este sistema operativo, la variedad de móviles con Windows Phone no es tan amplia como la que ofrece por ejemplo Android.

### **BlackBerry OS**

BlackBerry OS es un sistema operativo móvil desarrollado por BlackBerry para sus dispositivos. Su desarrollo se remonta a la aparición de las primeras PDAs en 1999. Su primera versión lanzada por RIM iba destinada a los buscapersoas de la firma canadiense. Este sistema operativo ofrecía correo electrónico, navegación web y sincronización con programas como Microsoft Exchange o Lotus Notes aparte de poder hacer las funciones usuales de un teléfono móvil.

Sin embargo, no fue hasta 2008 con el lanzamiento de BlackBerry OS 5, en parte gracias a la popularización del acceso a Internet desde el móvil y al lanzamiento de modelos asequibles pensados para un público joven, cuando dio un paso definitivo para afianzarse entre los principales sistemas operativos móviles.

La última versión desarrollada de este sistema operativo fue lanzada en el primer trimestre de 2013, bajo el nombre de BlackBerry 10. Como principal novedad de esta versión con respecto a las anteriores cabe destacar la incorporación de un teclado virtual. A pesar del poco tiempo que lleva en el mercado esta versión, la tienda BlackBerry World incluye aplicaciones compatibles con BlackBerry 10 desde su lanzamiento.

Al igual que ocurre en Android, los desarrolladores independientes pueden crear aplicaciones para BlackBerry sin tener que pagar por ello, ya que el SDK y el simulador son gratuitos. Sin embargo, en el caso de querer que dichas aplicaciones sean publicadas en la tienda oficial necesitan firmar digitalmente sus aplicaciones para asociarlas a una cuenta de desarrollador de RIM. El precio a pagar por dicha publicación es de 200\$ por cada 10 aplicaciones que se publiquen.

Por último, cabe destacar la escasa posibilidad de elección en cuanto a dispositivos que incorporen este sistema operativo. Aparte de los dispositivos de la propia marca, otras

marcas utilizan para algunos modelos el cliente de correo electrónico de BlackBerry. Entre esas marcas destacan Siemens, HTC y Sony Ericsson.

### **Symbian**

Symbian es un sistema operativo producto de la alianza de varias empresas de telefonía móvil en 1997. Entre dichas empresas se encuentran Nokia, Sony Ericsson, Psion, Samsung, Siemens, Arima, Benq, Fujitsu, Lenovo, LG, Motorola, Mitsubishi Electric, Panasonic, Sharp, etc.

Con el lanzamiento de la última versión, Nokia sustituye la denominación Symbian por Nokia Belle como parte de sus planes de unificación de marcas. Sin embargo, el sistema operativo Symbian sigue llamándose como tal ya que internamente y de cara a los desarrolladores sigue denominándose Symbian.

Las aplicaciones se distribuyen a través de la Ovi Store [OVISTORE], tienda oficial de descarga de aplicaciones, con un costo significativo para los desarrolladores de 1\$. A diferencia que ocurre con los desarrolladores de aplicaciones para iOS y Windows Phone, este pago se realiza una única vez, por lo que no tiene que pagar esta tarifa anualmente.

La Ovi Store cuenta con más de 50.000 aplicaciones para descargar y más de 6,5 millones de descargas diarias. A diferencia de versiones anteriores, en la actualidad la Ovi Store permite la actualización de las aplicaciones, ya que antes únicamente era posible descargarse la aplicación nuevamente y no sólo la actualización. Además, es posible realizar este proceso de actualización de una determinada aplicación vía PC y no únicamente vía OTA (Over The Air), para los casos en los que el usuario no disponga de paquete de datos o conexión WiFi disponible.

La mayoría de los dispositivos móviles que soportan el sistema operativo Symbian son de la compañía Nokia, aunque también se ha implementado en dispositivos de otros fabricantes entre los que destacan Siemens, Panasonic, Lenovo, LG, Samsung, Sony Mobile Communications, etc. Por todo esto se puede afirmar que este sistema operativo está disponible en una gran variedad de dispositivos móviles.

## 2.1.2 Elección del sistema operativo móvil

En la Figura 2.1 se pueden observar todas las características de cada uno de los sistemas operativos explicados en el apartado anterior, además de otras nuevas [GIR13].






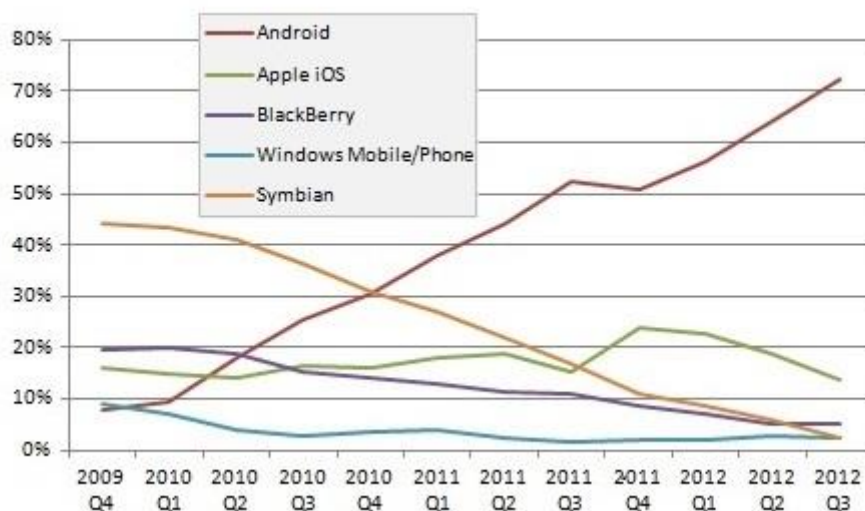
					
	Apple iOS 7	Android 4.2	Windows Phone 7	BlackBerry OS 7	Symbian 9.5
Compañía	Apple	Open Handset Alliance	Windows	RIM	Symbian Foundation
Núcleo del SO	Mac OS X	Linux	Windows CE	Mobile OS	Mobile OS
Familia CPU soportada	ARM	ARM, MIPS, Power, x86	ARM	ARM	ARM
Lenguaje de programación	Objective-C, C++	Java, C++	C#, muchos	Java	C++
Licencia de software	propietaria	software libre y abierto	Propietaria	propietaria	software libre
Año de lanzamiento	2007	2008	2010	1999	1997
Motor del navegador web	WebKit	WebKit	Pocket Internet Explorer	WebKit	WebKit
Soporte Flash	No	Sí	No	Si	Sí
HTML5	Sí	Sí	Sí	Sí	No
Tienda de aplicaciones	App Store	Google Play	Windows Marketplace	BlackBerry World	Ovi Store
Número de aplicaciones	400.000	300.000	50.000	30.000	50.000
Coste publicar	\$99 / año	\$25 una vez	\$99 / año	200 \$	\$1 una vez
Plataforma de desarrollo	Mac	Windows, Mac, Linux	Windows	Windows, Mac	Windows, Mac, Linux
Actualizaciones automáticas del S.O.	Sí	depende del fabricante	depende del fabricante	Sí	Sí
Soporte memoria externa	No	Sí	No	Sí	Sí
Fabricante único	Sí	No	No	Sí	No
Variedad de dispositivos	modelo único	muy alta	baja	baja	muy alta
Tipo de pantalla	capacitativa	capacitiva /resistiva	capacitativa	/resistiva capacitativa	capacitiva /resistiva
Aplicaciones nativas	Sí	Sí	No	No	Sí

Figura 2.1 Comparativa plataformas móviles

Otro aspecto fundamental a la hora de comparar las plataformas móviles es su cuota de mercado. En la Figura 2.2 se podrá observar la evolución del mercado de los sistemas operativos para móviles según el número de terminales vendidos hasta el tercer trimestre del año 2012 [GIR13].



**Figura 2.2 Cuota de mercado de los sistemas operativos para móviles**

Como se puede observar en la figura anterior, la plataforma Symbian de Nokia ha sufrido un importante descenso de ventas en los últimos años. Así mismo, cabe destacar el declive continuo de BlackBerry y como la plataforma de Windows parece que no despeg.

Las dos principales plataformas en lo que se refiere a la cuota de mercados son iOS de Apple y Android, aunque la diferencia entre ellas es bastante notable. Se observa como Apple tiene afianzada una cuota de mercado en torno al 15%. Por último, destaca el espectacular incremento exponencial que ha sufrido la plataforma Android, lo que le ha permitido alcanzar una cuota de mercado superior al 75% en menos de tres años.

Con todo esto se llega a la conclusión de que la aplicación se implementará para que funcione en dispositivos que tengan instalada la plataforma Android de Google. Una de las principales razones para tomar esta decisión ha sido la superior cuota de mercado existente con respecto a los otros sistemas operativos móviles estudiados, debido en gran medida a la amplia variedad de dispositivos que tienen instalado Android.

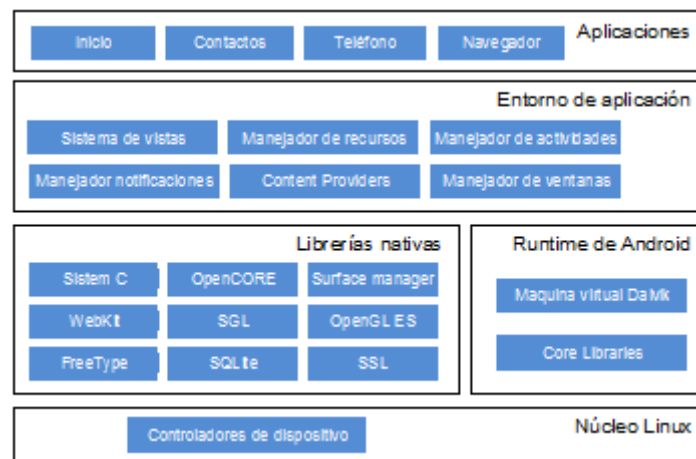
Otra de las razones fundamentales para tomar esta decisión es el coste de desarrollo, ya que únicamente hay que efectuar un único pago de 25\$ para obtener licencia de desarrollador y poder publicar un número ilimitado de aplicaciones, sin tener que seguir pagando para mantener dichas aplicaciones en la tienda oficial de Android al cabo de los años.

## 2.2 Android

### 2.2.1 Arquitectura de Android

Como se ha mencionado en el apartado anterior, Android es una plataforma para dispositivos móviles que contiene una pila de software donde se incluye un sistema operativo, middleware y aplicaciones básicas para el usuario.

En las siguientes líneas se dará una visión global por capas de cuál es la arquitectura empleada en Android. Como se puede observar en la Figura 2.3, está formada por varias capas, todas ellas basadas en software libre. Cada una de estas capas utiliza servicios ofrecidos por las anteriores, y ofrece a su vez los suyos propios a las capas de niveles superiores [GIR12] [ANDROIDEITY].



**Figura 2.3 Arquitectura de Android**

A continuación se explican detalladamente cada una de las capas de las que se compone la arquitectura de Android, así como de sus principales características:

#### Núcleo Linux

El núcleo del sistema operativo Android está basado en el kernel de Linux, versión 2.6. Esta capa del modelo actúa como una capa de abstracción entre el hardware y el resto de las capas de la arquitectura. El desarrollador no accede directamente a esta capa, sino que utiliza las librerías disponibles en capas superiores. Para cada elemento de hardware del teléfono existe un controlador (o driver) dentro del kernel que permite utilizarlo desde el software.

Esta capa también proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso y la pila de protocolos.

## **Runtime de Android**

Está basado en el concepto de máquina virtual utilizado en Java. Como ya se comentó anteriormente, los dispositivos tienen una serie de limitaciones como son la escasa memoria disponible y la limitada capacidad computacional comparada con los procesadores de los ordenadores de escritorio.

Debido a estas limitaciones de los dispositivos donde ha de ejecutarse Android, no fue posible utilizar una máquina virtual Java estándar. Google tomó la decisión de crear una nueva, la máquina virtual Dalvik [DALVIK], que respondiera mejor a estas limitaciones.

Algunas características de la máquina virtual Dalvik que facilitan esta optimización de recursos son:

- Ejecuta ficheros Dalvik ejecutables (.dex), que posee un formato optimizado para ahorrar memoria.
- Está basada en registros.
- Cada aplicación corre en su propio proceso Linux con su propia instancia de la máquina virtual Dalvik.
- Delega al kernel de Linux algunas funciones como threading y el manejo de la memoria a bajo nivel.

También se incluye en el Runtime de Android el “core libraries” con la mayoría de las librerías disponibles en el lenguaje Java.

## **Librerías nativas**

La siguiente capa que se sitúa justo sobre el kernel la componen las bibliotecas nativas de Android, también llamadas librerías. Están escritas en C o C++ y compiladas en el código nativo del procesador. El objetivo de estas librerías es proporcionar funcionalidad a las aplicaciones para tareas que se repiten con frecuencia, evitando de esta manera tener que codificarlas cada vez, y garantizando que se llevan a cabo de la forma más eficiente. Muchas de dichas librerías utilizan proyectos de código abierto.

A continuación se enumeran las principales librerías que se encuentran en esta capa:

**System C library:** se trata de una derivación de la librería BSD de C estándar (libc), adaptada para dispositivos embebidos basados en Linux.

**Media Framework:** librería pasada en PacketVideo’s OpenCORE; soporta codecs de reproducción y grabación de multitud de formatos de audio, vídeo e imágenes MPEG4, H.264, AAC, AMR, JPG y PNG.

**Surface Manager:** maneja el acceso al subsistema de representación gráfica en 2D y 3D.

**WebKit:** soporta un moderno navegador web utilizado en el navegador de Android y en la vista webview. Se trata de la misma librería que utiliza Google Chrome y Safari de Apple.

**SGL:** motor de gráfico 2D.

**Librerías 3D:** implementación basada en OpenGL ES 1.0 API. Las librerías utilizan el acelerador hardware 3D si está disponible, o el software altamente optimizado de proyección 3D.

**FreeType:** fuentes en bitmap y renderizado vectorial.

**SQLite:** potente y ligero motor de bases de datos relacionales disponible para todas las aplicaciones.

**SSL:** proporciona servicios de encriptación Secure Socket Layer.

### **Entorno de aplicación**

Proporciona una plataforma de desarrollo libre para aplicaciones con gran riqueza e innovaciones (sensores, localización, servicios, barra de notificaciones, etc.).

Esta capa ha sido diseñada para simplificar la reutilización de componentes. Las aplicaciones pueden publicar sus capacidades y otras pueden hacer uso de ellas, estando sujetas a las restricciones de seguridad. Este mecanismo permite a los usuarios remplazar componentes.

Una de las mayores fortalezas del entorno de aplicación de Android es que se aprovecha el lenguaje de programación Java. El SDK de Android no acaba de ofrecer todo lo disponible para su estándar del entorno de ejecución Java (JRE), pero es compatible con una fracción muy significativa de la misma.

Los servicios más importantes que incluye son:

**Views:** Este servicio representa el elemento básico para los componentes de interfaz de usuario. Una *view* ocupa un área rectangular en la pantalla y es la responsable del dibujo y de la gestión de eventos. View es la clase base para los *widgets*, que se utilizan para crear componentes interactivos de interfaz de usuario (botones, campos de texto, etc.) [VIEW].

**Resource Manager:** Con esta librería se pueden gestionar todos los elementos que forman parte de la aplicación y que están fuera del código, tales como cadenas de texto traducidas a diferentes idiomas, imágenes, sonidos, *layouts*, etc.

**Activity Manager:** Se encarga de administrar la pila de actividades de la aplicación. Maneja el ciclo de vida de las aplicaciones y proporcionan un sistema de navegación entre ellas.

**Notification Manager:** Engloba los servicios para notificar al usuario cuando algo requiera su atención mostrando alertas en la barra de estado. Permite activar el vibrador o utilizar los LEDs del teléfono en caso de tenerlos. En definitiva, permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.

**Content Providers:** Esta librería crea una capa que encapsula los datos que se comparten entre aplicaciones (como los contactos de la agenda) para tener control sobre cómo se accede a la información.

### **Aplicaciones**

Este nivel está formado por el conjunto de aplicaciones instaladas en una máquina Android. Todas las aplicaciones han de ser ejecutadas en la máquina virtual Dalvik para garantizar la seguridad del sistema.

Normalmente, las aplicaciones Android están escritas en Java. Para desarrollar aplicaciones en Java se puede utilizar el Android SDK. Existe otra opción consistente en desarrollar aplicaciones utilizando C/C++. Para esta opción se puede utilizar el Android NDK (Native Development Kit).

## **2.2.2 Versiones de Android y niveles de API**

Antes de empezar un proyecto en Android, es determinante elegir la versión del sistema para la que se desea realizar la aplicación. Es muy importante observar que hay clases y métodos que están disponibles a partir de una versión [GIR13].

A continuación se describen las plataformas lanzadas hasta la fecha con una breve descripción de las novedades introducidas [GR13] [ANDROIDZONE]. Las plataformas se identifican de tres formas alternativas: versión, nivel de API y nombre. El nivel de API corresponde a números enteros comenzando desde el número 1.

Como se ha comentado anteriormente, para los nombres de las versiones se han elegido nombres de postres por orden alfabético, a excepción de las dos primeras versiones, que hubieran correspondido a las letras A y B y no recibieron nombre.

### **Android 1.0 Nivel de API 1 (septiembre 2008)**

Se trata de la primera versión de Android. En ella se incluyó un navegador web, soporte para mensajes de texto SMS y MMS y una aplicación básica para tomar fotos. No tiene sentido desarrollar aplicaciones para esta primera versión, ya que nunca se utilizó comercialmente.



### **Android 1.1 Nivel de API 2 (febrero 2009)**

Se trata de la primera versión que se utilizó comercialmente. Es la opción a escoger si se quiere desarrollar una aplicación que sea compatible con todos los dispositivos Android, aunque apenas se añadieron funcionalidades con respecto a la primera versión.

### **Android 1.5 Nivel de API 3 (abril 2009, Cupcake)**

Se trata de la primera actualización importante del sistema operativo. Es la primera versión con algún usuario (aunque apenas la usa un 0,1% en enero de 2013). En ella se incorpora la funcionalidad de teclado en pantalla con predicción de texto, por lo que los dispositivos no tienen que tener un teclado físico. Además, se añade la posibilidad de insertar *widgets*, se incorpora la auto-sincronización y el soporte para *bluetooth* estéreo y las transiciones animadas entre pantallas.

### **Android 1.6 Nivel de API 4 (diciembre 2009, Donut)**

Esta nueva versión proporciona al usuario la posibilidad de realizar una búsqueda avanza más rápida en todo el dispositivo desde la pantalla de inicio del teléfono, integrando tanto la búsqueda en Internet como la búsqueda local (contactos, aplicaciones, calendario, etc.). Así mismo, se facilita que las aplicaciones puedan utilizar distintos tamaños y resoluciones de pantalla. Se realizan mejoras en el tiempo de carga de la aplicación de la cámara. Proporciona soporte para CDMA/EVDO y VPNs. Por último, cabe destacar como principal novedad de esta versión la incorporación de la síntesis de texto a voz.

### **Android 2.0.x Nivel de API 5 (octubre 2009, Éclair)**

Como novedades para esta versión cabe destacar la mejora en la gestión de contactos, permitiendo agregar múltiples cuentas al dispositivo para sincronización de contactos y correo. Así mismo, se incorporan nuevas características para la cámara, como soporte de flash, efecto de colores, zoom digital, etc. En lo que se refiere a la interfaz de usuario del navegador, en esta versión cuenta con soporte para HTML5. También cabe destacar que se incorpora un API para manejar Bluetooth 2.1, así como una optimización en lo referente a la velocidad de hardware. Por último, se incorpora un diccionario inteligente en el teclado virtual del dispositivo que mejora la velocidad de mecanografía al incluir sugerencias o nombres de contactos.

### **Android 2.1 Nivel de API 7 (enero 2010, Éclair)**

Se considera una actualización menor, por lo que mantuvo el nombre de su antecesor. Como novedad principal de esta nueva versión destaca la posibilidad de introducir un campo de texto sin necesidad de utilizar el teclado al incluir reconocimiento de voz. También incorpora mecanismos para modificar la resolución de la pantalla, administrar la

configuración de la caché de aplicaciones, y almacenamiento web. Por último, en el paquete WebKit se incluyen nuevos métodos para manipular bases de datos.

### **Android 2.2.x Nivel de API 8 (mayo 2010, Froyo)**

Como característica principal destaca el notable incremento de la velocidad del sistema, tanto en sus aplicaciones como en la navegación de Internet, siendo de dos a cinco veces más rápida la ejecución del código de la CPU con respecto a la versión anterior.

También se añaden varias mejoras relacionadas con el navegador Web, como el soporte de Adobe Flash 10.1 y la incorporación del motor JavaScript V8 utilizado en Chrome.

Gracias al desarrollo de aplicaciones se introducen importantes novedades en esta versión como la de dar la opción de mover las aplicaciones a la tarjeta microSD. Así mismo, también se introduce una notable mejora en lo que se refiere a la copia de seguridad de los datos de las aplicaciones, ya que éstas son actualizadas de forma automática cuando aparece una nueva versión. Por último, se facilita que las aplicaciones interaccionen con el reconocimiento de voz y que terceras partes proporcionen nuevos motores de reconocimiento.

### **Android 2.3.x Nivel de API 9 (diciembre 2010, Gingerbread)**

Debido al éxito de Android en las tabletas, esta versión soporta mayores tamaños de pantalla con el soporte para resoluciones WXGA y superiores. Así mismo, incorpora una nueva interfaz de usuario con un diseño actualizado, en el que destaca un teclado en pantalla con capacidad multitáctil y la función para corregir palabras ya ingresadas mediante la voz.

Así mismo, incluye soporte nativo para varias cámaras, pensando en la segunda cámara usada en videoconferencia. Debido a la inclusión de esta segunda cámara, se implementa la función de reconocimiento facial para identificar al usuario del terminal.

Entre otras novedades destacan el soporte nativo para telefonía sobre Internet VoIP/SIP, el soporte para reproducción de vídeo WebM/VP8 y codificación de audio AAC, el soporte para la tecnología NFC, las facilidades en el audio, el soporte nativo para más sensores y un gestor de descargas para las descargas largas.

### **Android 3.x Nivel de API 11 (febrero 2011, Honeycomb)**

Esta versión es lanzada para mejorar la experiencia de Android en las nuevas tabletas, siendo optimizada para dispositivos con pantallas grandes. La nueva interfaz que incorpora esta versión se pone a disposición de todas las aplicaciones, incluso las construidas para versiones anteriores de la plataforma. Se simplifica la función multitarea, mostrando una lista con los iconos de las aplicaciones recientes, para que el usuario pueda acceder a ellas.

También se realizan notables evoluciones en lo que se refiere al navegador, ya que se incluye la característica de auto-completado al ingresar texto y el modo incógnito, que permite la navegación de forma anónima con el navegador web.

Se añaden varias mejoras multimedia, como listas de reproducción M3U a través de HTTP Live Streaming, soporte a la protección de derechos musicales (DRM) y soporte para la transferencia de archivos multimedia a través de USB con los protocolos MTP y PTP.

A pesar de que la nueva interfaz gráfica es optimizada para tabletas, esta versión de Android es compatible con las aplicaciones creadas para versiones anteriores.

En mayo de 2011 se lanza la versión 3.1, que permite manejar dispositivos conectados por USB. También se instaure un protocolo de transferencia de fotos y vídeos (PTP/MTP) y de tiempo real (RTP).

Un mes después aparece la versión 3.2, con la que se realizan optimizaciones para distintos tipos de tablets. También se permite la sincronización multimedia desde un medio de almacenamiento externo, como una tarjeta SD.

#### **Android 4.0.x Nivel de API 14 (octubre 2011, Ice Cream Sandwich)**

La principal característica de esta actualización es que integra sus dos versiones anteriores (2.x para teléfonos y 3.x para tabletas) en una sola, siendo compatible con cualquier tipo de dispositivo.

Sin embargo, también cabe destacar otras características interesantes como la de una nueva interfaz de usuario totalmente renovada. Se introduce la posibilidad de acceder a las aplicaciones directamente desde la pantalla de bloqueo. Además, se incluye un nuevo API de reconocedor facial que permite entre otras cosas desbloquear el dispositivo. También se mejora en el reconocimiento de voz.

Aparece un nuevo gestor de tráfico de datos por Internet, donde el usuario puede gestionar de forma nativa el consumo de datos de todo el equipo y definir los límites para evitar exceder los límites de dicho consumo.

Se establece Google Chrome como navegador por defecto, permitiendo con ello que el usuario pueda abrir hasta 15 pestañas y realizar la sincronización automática con los marcadores de la versión de escritorio.

En la versión 4.0.3 se introducen ligeras mejoras en algunas APIs incluyendo el de redes sociales, calendario, revisor ortográfico, texto a voz y bases de datos entre otros.

#### **Android 4.1 Nivel de API 16 (julio 2012, Jelly Bean)**

Se trata de la última actualización importante del sistema operativo hasta la fecha. El objetivo primordial de esta versión es el de mejorar la estabilidad, funcionalidad y rendimiento de la interfaz de usuario, para lo cual se implementa el núcleo de Linux 3.0.31.

Se introducen varias mejoras en Google Search. Se potencia la búsqueda por voz con resultados en forma de ficha. Para mejorar la seguridad las aplicaciones son cifradas. También se permiten actualizaciones parciales de las aplicaciones.

Finalmente, otra de las mejoras importantes que presenta esta versión, es la de no requerir conexión a Internet para poder utilizar la entrada por voz. Esto es posible gracias que el intérprete se encuentra dentro del dispositivo y al teclado predictivo que incorpora, capaz de adivinar la próxima palabra que el usuario va a escribir.

#### **Android 4.2.x Nivel de API 17 (noviembre 2012, Jelly Bean)**

Una de las novedades más importantes de esta versión es la incorporación de soporte multiusuario para tabletas. Cada usuario puede tener su espacio independiente con sus aplicaciones y su correspondiente configuración. Permite el cambio de cuenta sin necesidad de cerrar sesión.

Se incorpora un nuevo teclado predictivo deslizante al estilo Swype, que permite al usuario escribir sin tener que levantar el dedo de la pantalla. Por último, se incluye una versión mejorada de Google Now, y un rediseño en lo que se refiere a la interfaz de la cámara, ocultando las barras laterales y los botones, por lo que la imagen ocupa la pantalla por completo.

#### **Android 4.3.x Nivel de API 18 (mayo 2013, Jelly Bean)**

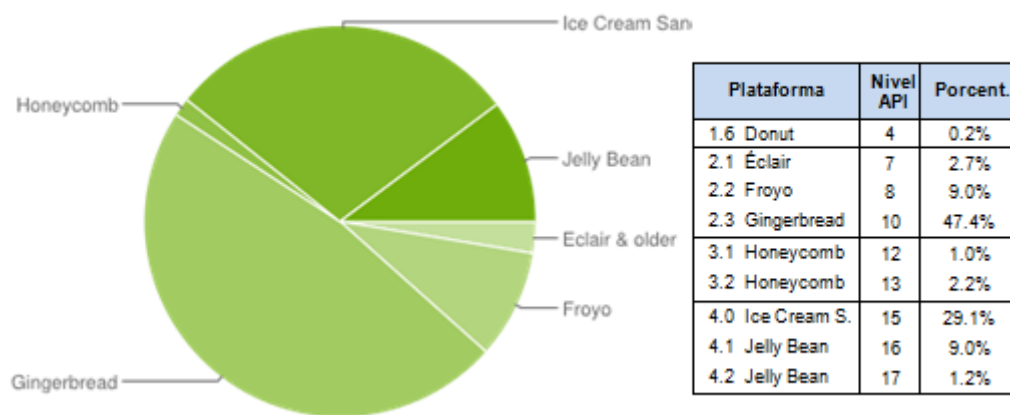
Se trata de la última versión publicada hasta la fecha. En ella se incorpora un administrador de permisos, con el que se puede desactivar algunos de los permisos de las aplicaciones. En lo que se refiere a los gráficos, da soporte oficial a OpenGL ES 3.0. También se incorpora la posibilidad de revocar el acceso a la depuración de todos los ordenadores a los que se ha autorizado de manera global.

Se añade un listado de aplicaciones inhabilitadas en el que se puede visualizar todas las aplicaciones que han sido desactivadas. Permite la activación de un control paternal, con el que se permite decidir a qué aplicaciones y/o contenidos puede acceder un determinado usuario.

### 2.2.3 Android 2.3 vs. Android 4.0

A la hora de seleccionar la plataforma de desarrollo hay que consultar si la aplicación a desarrollar necesita alguna característica especial que sólo esté disponible a partir de una determinada versión, ya que todos los usuarios con versiones inferiores a la seleccionada no podrán hacer uso de la aplicación.

Por lo tanto, es recomendable seleccionar la menor versión posible que la aplicación pueda soportar. La Figura 2.4 muestra los porcentajes de utilización de las distintas plataformas desarrolladas por Android hasta la fecha [GIR13].



**Figura 2.4 Utilización de dispositivos Android según plataforma instalada**

De la figura anterior se puede destacar cómo las dos principales versiones de Android en la actualidad son la versión 2.3 de Android, denominada Gingerbread, y la versión 4.0, denominada Ice Cream Sandwich.

Debido a que el requisito fundamental de la aplicación a desarrollar es su acceso oral, la versión a elegir tiene que tener ya instalada la librería tanto de síntesis de texto a voz como de reconocimiento de voz. Como se ha comentado a la hora de explicar en detalle cada una de las versiones, la síntesis de texto a voz apareció en la versión 1.6 de Android y el reconocimiento de voz en la versión 2.1 por lo que las dos versiones con mayor porcentaje de utilización siguen siendo válidas, pues ambas son posteriores a las mencionadas.

No obstante, es recomendable seleccionar la menor versión posible, pues aquellas aplicaciones que son aptas para funcionar en Android 4.0 también lo son para hacerlo en versiones anteriores. Por todo esto, la versión elegida para realizar la aplicación es la versión 2.3 de Android correspondiente con el nivel de API 10.

## 2.3 Los sistemas de diálogo

Los sistemas de diálogo basados en procesamiento del habla son sistemas informáticos que reciben como entrada frases del lenguaje natural expresadas de forma oral y generan como salida frases del lenguaje natural expresadas asimismo de forma oral [LOP05].

La finalidad de estos sistemas es emular el comportamiento inteligente de un ser humano en un diálogo con otra persona, con el objetivo de que se realice una cierta tarea [GRI07].

Un sistema de diálogo ideal reconocería el habla espontánea, comprendería enunciados sin restricciones de contenido, proporcionaría respuestas con sentido, gramaticalmente bien formadas y pragmáticamente adecuadas, respondería con voz completamente natural y sería multimodal [LLI06].

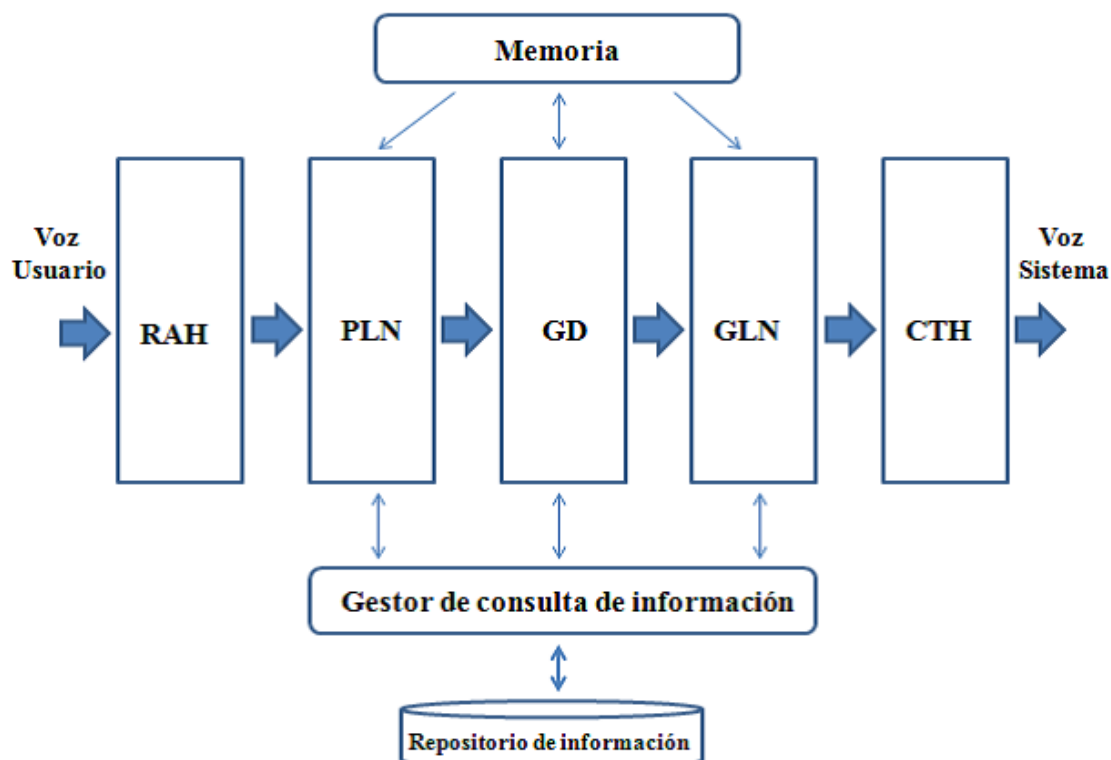
Actualmente, los sistemas de diálogo están limitados debido a encontrarse sujetos a las restricciones del reconocimiento automático del habla, estar la comprensión y respuesta limitados a dominios específicos, necesitar estrategias de verificación, estar condicionados por la naturalidad del habla sintetizada y existir el problema del diálogo que surge de forma espontánea.

El abanico de posibilidades en cuanto a entornos y tareas en los que se pueden aplicar estos sistemas es muy amplio, ya que permiten el acceso a servicios y control de máquinas por vía telefónica. La posibilidad de convertir cualquier línea de la red telefónica en un punto de acceso a los servicios les proporciona un gran valor económico y social: banca telefónica, centros de atención de llamadas, etc.

### 2.3.1 Arquitectura de un sistema de diálogo

El esquema utilizado para el desarrollo de sistemas de diálogo suele englobar una serie de módulos genéricos que deben coordinarse para responder a los requerimientos del usuario [LLI06] [HUR+05].

La Figura 2.5 muestra la arquitectura de un sistema de diálogo basado en el procesamiento del habla.



**Figura 2.5** Arquitectura modular de un sistema de diálogo

La descripción de los módulos y sus principales características se explican a continuación:

**Módulo de Reconocimiento Automático del Habla (RAH):** Procesa la voz de usuario y la transforma en la secuencia de palabras reconocidas con mayor probabilidad.

**Módulo de Procesamiento del Lenguaje Natural (PLN):** Extrae el significado de las palabras reconocidas en el módulo anterior, obteniendo la representación semántica (significado) de la frase y expresándolo en términos de un lenguaje especificado para la tarea.

**Gestor del diálogo (GD):** Decide qué paso debe dar el sistema tras cada intervención del usuario. Puede decirse que este es el módulo fundamental del sistema, pues su finalidad es lograr que la interacción con el usuario sea lo más cómoda e “inteligente” posible. Para ello, se basa en la interpretación semántica de la petición del usuario, la historia del proceso de diálogo, la información disponible en ese punto, el estado actual del sistema, la información obtenida de la base de datos, la estrategia definida, etc.

**Módulo de Generación del Lenguaje Natural (GLN):** Tiene como función la generación de una frase, gramaticalmente correcta y en un lenguaje lo más cercano posible al lenguaje natural, que transmita el mensaje generado por el gestor de diálogo.

**Módulo de Conversión Texto Habla (CTH):** Transforma la frase de respuesta en señal de audio.

**Módulo de memoria:** Almacena las representaciones semánticas obtenidas a lo largo de la interacción así como las frases previamente generadas por el sistema, proporcionando esta información histórica a los módulos de procesamiento del lenguaje natural, gestor del diálogo y generación del lenguaje natural. De esta forma, el sistema puede resolver las referencias anafóricas existentes en las frases pronunciadas por los usuarios, puede conocer qué frases ha expresado el usuario previamente, y puede utilizar información contextual (mediante el uso de anáforas y elipsis) durante la generación de las frases.

**Módulo gestor de consulta de información de la aplicación:** Se encarga de generar las consultas necesarias al repositorio de información, procesarlas y proporcionar la información obtenida al módulo gestor del diálogo.

**Repositorio de información:** Se trata de bases de datos o páginas web donde se almacena la información necesaria para el sistema de diálogo y que será consultada por el gestor de consulta de información.



# Capítulo 3

## Descripción general del sistema

En este capítulo se describen las características fundamentales del conjunto de módulos que conforman el sistema desarrollado: funcionalidad, arquitectura y esquema general de bloques. A continuación se analizan las distintas herramientas utilizadas así como los recursos necesarios para su desarrollo. Finalmente, se incluyen las operaciones más generales y codificaciones especiales que se han tenido en cuenta a lo largo de la implementación.

### 3.1 Presentación del sistema

La aplicación desarrollada para este Trabajo Fin de Grado ofrece al usuario una serie de funcionalidades que se enumeran a continuación. Dichas funcionalidades podrán ser realizadas tanto por voz como mediante el teclado virtual del dispositivo.

- Registrarse/validarse en la aplicación.
- Visualizar los contactos de la agenda.
- Realizar una llamada telefónica a un contacto existente.
- Añadir un nuevo contacto a la agenda telefónica.
- Modificar un contacto existente.
- Eliminar un contacto de la agenda.
- Visualizar la bandeja de mensajes.
- Recibir mensajes de otros usuarios.
- Enviar un nuevo mensaje.
- Obtener una lista ordena de los contactos en función a las veces que el usuario se ha puesto en contacto con ellos.
- Realizar una serie de cambios en la configuración de la aplicación.

Cada uno de estos servicios se explica más detalladamente en el siguiente capítulo. La Figura 3.1 representa gráficamente las funcionalidades anteriormente citadas.



**Figura 3.1 Servicios ofrecidos por la aplicación**

La arquitectura de la aplicación desarrollada varía dependiendo de si el usuario realiza la acción desea mediante voz o mediante el teclado virtual del dispositivo.

La interacción con el sistema empieza cuando el usuario se registra o se valida en la aplicación. Una vez realizado este procedimiento, todas las demás acciones, a excepción del cambio de contraseña, podrán ser realizadas mediante la inserción de texto o voz.

La principal diferencia, en lo que se refiere a la arquitectura de la aplicación, entre realizar una acción mediante voz o texto, radica en el hecho de que si se hace uso del reconocimiento de voz, y el dispositivo móvil tiene un nivel de API inferior a 17 (Android 4.2), la aplicación deberá de tener acceso a Internet para poder procesar la entrada de audio y generar el texto correspondiente.

Después de que el usuario haya especificado la acción que desea realizar, el procesamiento de la información por parte de la aplicación en ambos casos será equivalente. Dependiendo de la acción demandada por el usuario, la aplicación accederá a la agenda del dispositivo para cargar los contactos, o realizar una serie de acciones sobre la misma, o bien a la base de datos de la aplicación para cargar información relativa a los contactos más frecuentes o a los mensajes enviados y/o recibidos.

## **3.2 Herramientas del sistema**

A continuación se especifican las herramientas empleadas para poder llevar a cabo el desarrollo del Trabajo Fin de Grado. Estas herramientas utilizadas han sido de open source; es decir, no ha supuesto coste alguno el utilizarlas ya que no era necesario realizar ningún pago para obtener las licencias de las mismas.

### **3.2.1 Eclipse**

Se trata de un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma. Para el desarrollo de este Trabajo Fin de Grado se ha utilizado la distribución más reciente en la actualidad, Eclipse ‘Kepler’.

Tal y como se ha comentado anteriormente, al tratarse de una herramienta de código abierto no ha sido necesario desembolso económico alguno para su utilización. El recurso está disponible en la página oficial de Eclipse [ECLIPSE].

No obstante, para el desarrollo de aplicaciones en Android, es necesario instalarse un determinado plugin. Google provee principalmente dos plugins para desarrolladores en Android; el primero corresponde al desarrollo de aplicaciones web, App – Engine, y el segundo es el correspondiente con el desarrollo de aplicaciones en Android, que es el necesario para el desarrollo de este Trabajo Fin de Grado.

Todos los pasos necesarios para la correcta instalación de dicho plugin se detallan en la página oficial de Google [PLUGIN].

#### **3.2.1.1 SDK Android**

Después de realizar los pasos anteriormente detallados, ya es posible desarrollar aplicaciones en Android con Eclipse. Sin embargo, para poder implementarlas es necesario descargarse el SDK de Android [SDK].

Debido a que el proyecto está compilado con la versión de Android 4.0.3, es necesario tener descargado el paquete de dicha versión o el de una versión superior.

Una vez instalado el SDK de Android, se pueden realizar pruebas sobre dispositivos Android emulados. Por lo tanto, se puede asumir que un escenario mínimo para el desarrollo de este tipo de aplicaciones sería la integración de Eclipse y SDK Android.

### 3.2.1.2 Sistema de ficheros

Como se puede observar en la Figura 3.2, la arquitectura de archivos de la aplicación se divide en cuatro paquetes, todos ellos con una nomenclatura constante '*android.tfg*'. A partir de ahí su nombre determina su función en la aplicación.



**Figura 3.2 Sistema de ficheros de la aplicación desarrollada**

El primero de los paquetes se denomina 'activity'. En él se encuentra la parte principal de la aplicación, las Actividades. Estas Actividades implementan el funcionamiento que debe tener cada una de las interfaces de la aplicación. Por cada interfaz, existe su correspondiente Actividad.

El segundo de los paquetes, denominado 'object' se encarga de crear los objetos que van a necesitar implementar las Actividades, como pueden ser los objetos SMS y Contact.

En el tercer paquete se ubica el Broadcast Receiver encargado de capturar todos los mensajes entrantes que reciba el dispositivo móvil.

El último de los paquetes, denominado 'utils', contiene aquellos ficheros Java necesarios para la creación de la base de datos y de los cifradores con los que se cifra y descifra la misma.

### 3.2.2 SQLite

Para el correcto funcionamiento de la aplicación, es necesaria la creación de una base de datos en la que se almacene información sensible, tanto del propio usuario de la aplicación como de sus contactos. Para ello, se ha hecho uso de la base de datos que viene integrada en la librería de Android, SQLite, cuyo lenguaje es muy similar al de MySQL.

SQLite es un motor de datos de dominio público que implementa una pequeña librería programada en lenguaje C. Al tener una arquitectura cliente/servidor independiente, la latencia en cuanto al acceso a las bases de datos es menor en comparación con otros motores de bases de datos convencionales [BUR12].

Debido a las siguientes ventajas que se enumeran a continuación [PREZI], se ha estimado oportuno que SQLite es la opción ideal para en lo referente a la base de datos para la aplicación a desarrollar para este Trabajo Fin de Grado:

- Es pequeño, la versión actual no ocupa más de 360 KB, lo que resulta perfecto para la capacidad de memoria de un dispositivo Android.
- Realiza operaciones de manera eficiente, teniendo mejor rendimiento que otros motores de bases de datos como MySQL y PostgreSQL.
- No necesita un proceso separado funcionando como servidor ya que lee y escribe directamente sobre archivos que se encuentran en el disco duro.
- Emplea registros de tamaño variable de tal forma que utiliza únicamente el espacio en disco que realmente es necesario en cada momento.
- Se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.

Por último, cabe destacar que, al igual que con Eclipse, SQLite es de dominio público, por lo que su utilización para la realización de la aplicación no ha supuesto coste adicional.

## **3.3 Recursos necesarios para el sistema**

En este apartado se especifican los recursos necesarios para el correcto funcionamiento del sistema, haciendo una breve descripción de cada uno de ellos así como de su uso dentro de la aplicación desarrollada.

### **3.3.1 Google Voice Recognizer**

Este sistema de reconocimiento de voz se encuentra integrado en el sistema operativo Android desde el nivel de API 8. Se trata de un servicio gratuito que requiere conexión a Internet para que se produzca el reconocimiento de la frecuencia de voz y por tanto del mandato vocalizado. Este reconocedor de voz se activará cada vez que el usuario pulse sobre un determinado botón. Una vez iniciado el servicio, éste seguirá activo hasta que deje de recibir una entrada de audio, por lo que la duración de este servicio será variable dependiendo del tiempo que el usuario esté hablando.

En caso de que no reciba una entrada de audio, las funciones correspondientes a este servicio no se ejecutarán, evitando así que se pueda producir alguna excepción no controlada.

### **3.3.2 Text To Speech**

Al igual que ocurre con el recurso anteriormente explicado, se trata de un servicio integrado en el propio sistema operativo, accesible desde la propia API de Android. Este servicio es el encargado de reproducir en voz alta un texto previamente introducido.

Este sistema de reproducción de texto en forma sonora es el utilizado para hacer que la aplicación interactúe con el usuario, por lo tanto es uno de los recursos más utilizados de la aplicación, ya que ésta está constantemente comunicándose con el usuario. En el siguiente capítulo se comentará en qué casos se implementará este servicio.

En la Figura 3.3 se muestra un ejemplo de cómo se inicializa y se utiliza este servicio en la aplicación desarrollada. Para poder integrar este servicio en alguna de las actividades de las que se compone la aplicación, es necesario llamar a la librería ‘*android.speech.tts.TextToSpeech*’ de Android.

```
TextToSpeech tts = new TextToSpeech(getApplicationContext(), this);

private void speakOut(String text) {
    tts.speak(text, TextToSpeech.QUEUE_FLUSH, null);
}
```

**Figura 3.3 Inicialización y uso del conversor de texto a voz**

### 3.3.3 Broadcast Receiver

Un receptor de anuncios, *Broadcast Receiver*, es un recurso que recibe y reacciona ante anuncios globales que se producen en dispositivos móviles cuyo sistema operativo es Android. Se utiliza para mostrar notificaciones de los eventos como pueden ser el agotamiento de la batería o el bloqueo de pantalla, aunque las aplicaciones también pueden lanzar anuncios de este tipo.

A pesar de que los Broadcast Receiver no tienen interfaz de usuario pueden iniciar una actividad o crear una notificación para informar al usuario.

El ciclo de vida de un receptor de anuncios es muy sencillo ya que únicamente dispone de un método *onReceive()* que será llamado por el sistema cuando se produzca el evento en cuestión. Una característica de este tipo de recurso es que no hace falta que se esté ejecutando la aplicación en la que se define el Broadcast Receiver para que este se active.

En el caso de la aplicación desarrollada para el Trabajo Fin de Grado, se ha hecho uso de un Broadcast Receiver para capturar aquellos mensajes que recibe el usuario. De esta manera, cuando el usuario reciba un mensaje se activará este Broadcast y procesará el mensaje recibido como corresponda. Se explicará más detalladamente el funcionamiento de este Broadcast Receiver en el capítulo ‘4.11 Módulo Recibir Mensaje’.

Tal y como se ha comentado en el apartado ‘3.2.1.2 Sistema de ficheros’ la clase en la que está incluida el Broadcast estará en el paquete ‘*android.tfg.receiver*’.

## 3.4 Implementación de las operaciones generales

### 3.4.1 Gramáticas

Para todos los diálogos que mantienen la aplicación y el usuario, existe una serie de palabras reservadas que implicarán acciones que la aplicación llevará a cabo. Dependiendo de la interfaz en la que el usuario se encuentre estas palabras variarán. En el siguiente capítulo se explicarán más detalladamente qué palabras están reservadas para cada uno de los módulos en los que se divide la aplicación.

Así mismo, a la hora de introducir un número telefónico mediante el reconocedor de voz anteriormente explicado, se llevará a cabo una transformación de dicho número de su forma escrita a su forma numérica.

### 3.4.2 Gestión de Base de Datos

Tal y como se ha comentado anteriormente, para el correcto funcionamiento de la aplicación se ha hecho uso del motor de bases de datos SQLite para crear la base de datos que almacene toda la información que la aplicación pueda necesitar. Todo lo relativo a la base de datos se encuentra en la clase SQLite.java, dentro del paquete '*android.tfg.utils*'.

Para asegurar la confidencialidad de la información que en ella se almacena, se ha optado por cifrar aquellos datos que se consideren sensibles.

En la Figura 3.4 se muestra cómo se generan las distintas tablas, así como los distintos campos de los que se componen cada una de ellas.

```
public void onCreate(SQLiteDatabase db) {  
    db.execSQL("CREATE TABLE user (nick TEXT, token TEXT, key TEXT, audio TEXT)");  
    db.execSQL("CREATE TABLE contacts (idContact TEXT, name TEXT, amount TEXT)");  
    db.execSQL("CREATE TABLE messages (idContact TEXT, name TEXT, origin TEXT, " +  
        "message TEXT, time TEXT, isRead TEXT)");  
}
```

**Figura 3.4 Base de datos de la aplicación desarrollada**

A continuación se detallan los campos de cada una de las tablas así como su finalidad para aplicación desarrollada.



### **Tabla ‘user’**

En esta tabla se almacena toda la información relativa al usuario de la aplicación. Debido a que no se contempla que varios usuarios estén registrados en la aplicación para un mismo dispositivo móvil, esta tabla tendrá siempre una fila.

- Nick: Este campo contiene el nombre que el usuario ha elegido para registrarse. Cuando la aplicación se comunique con el usuario mediante el sistema de reproducción de texto en voz, se dirigirá al usuario con este nombre. De esta manera se intenta simular una conversación lo más humanizada posible. Este campo únicamente se actualiza cuando el usuario se registra en la aplicación.
- Token: Se trata de un número aleatorio de 16 bytes que se utiliza para cifrar la contraseña que el usuario ha introducido a la hora de registrarse. Con esta variable se intenta otorgar una mayor seguridad al cifrado de la base de datos. Este número se genera de forma transparente para el usuario cuando éste se registra en la aplicación, y ya no se actualiza su valor.
- Key: En este campo se guarda la contraseña que el usuario introdujo cuando se dio de alta en la aplicación. Esta variable servirá también como clave para cifrar la base de datos. A diferencia de los campos anteriores, éste sí podrá ver afectado su valor, ya que el usuario tendrá la opción de cambiar la contraseña siempre que lo desee.
- Audio: Este campo determina si el usuario tiene habilitada la opción de que la aplicación se comunique con él o no. Tomará valor true o false.

Se ha considerado que los campos *nick*, *token* y *key* almacenan información sensible, por lo que se ha cifrado su valor. Para los campos *nick* y *token* se ha utilizado como clave la contraseña con la que el usuario se valida en la aplicación. Esta contraseña a su vez es cifrada con el número aleatorio (*token*) cifrado. De esta manera se obtiene un cifrado sólido difícil de ser roto.

### **Tabla ‘contacts’**

Esta tabla se utiliza para saber la cantidad de veces que el usuario se ha comunicado, bien vía mensaje de texto o bien mediante una llamada telefónica, con cada uno de los contactos de su agenda. A diferencia de la tabla anterior, la cantidad de filas de ésta puede ser variable, pudiendo ser nula o pudiendo ser como máximo el número de contactos que el usuario tiene en su agenda.

- idContact: Almacena el número de teléfono del contacto al que el usuario le ha enviado un mensaje o le ha llamado. Debido a que el número telefónico es único por cada contacto, se considerará clave primaria.

- Name: Este campo contiene el nombre con el que el usuario tiene asociado el número telefónico que se almacena en el campo anterior. Este nombre se obtendrá de la agenda de contactos.
- Amount: Este último campo guarda el número de veces que el usuario de la aplicación ha enviado un mensaje o realizado una llamada con el contacto en cuestión.

Para esta tabla, todos los campos se consideran información sensible, por lo que todos han sido cifrados utilizando como clave la contraseña que se almacena en la tabla *user* con su valor en claro (antes de que éste se cifrara).

A pesar de que el campo *amount* almacena un valor numérico, se ha decidido que éste se almacene como si de una cadena de caracteres se tratase. De esta manera se facilita su cifrado.

### **Tabla ‘messages’**

En esta última tabla se almacenará toda la información relativa a los mensajes de texto, tanto a los enviados como a los recibidos. A continuación se describen los campos que la componen.

- idContact: Al igual que en la tabla anterior, en este campo se almacena el número de teléfono del contacto al que el usuario le ha enviado un mensaje o del que el usuario ha recibido un mensaje. Debido a que el número telefónico es único por cada contacto, se considerará clave primaria.
- Name: Este campo guarda el nombre con el que el usuario tiene asociado el número telefónico que se almacena en el campo anterior. Este nombre se obtendrá de la agenda de contactos.
- Origin: Esta columna sirve para determinar si el mensaje es enviado por el usuario, tomará valor 2, o ha sido recibido, tomará valor 1.
- Message: En este campo se almacena el mensaje propiamente dicho.
- Time: Este campo contiene la hora a la que el mensaje se envió/recibió. También guardará la fecha con el día mes y año.
- isRead: Esta última columna sirve para determinar si el mensaje ya ha sido leído por el usuario de la aplicación o si por el contrario todavía sigue sin leerse.

Para esta última tabla se considerará información sensible todas las columnas a excepción de la última. Nuevamente la clave a utilizar para cifrar la información será la contraseña introducida por el usuario.

### 3.4.2.1 Cifrado y descifrado de la base de datos

Como se ha comentado al principio de este documento, los usuarios potenciales de la aplicación desarrollada son usuarios inexpertos en las nuevas tecnologías. Por este motivo, es importante otorgarle robustez y seguridad a la aplicación, ya que se asume que los usuarios carecen de conocimientos de seguridad.

El modo de cifrado escogido ha sido un cifrado de clave simétrica, también denominado cifrado de clave secreta. Este método emplea la misma clave para cifrar y descifrar, por lo que gran parte de la seguridad radica en la fuerza de la contraseña.

Debido a este motivo se ha desarrollado un algoritmo de derivación de clave con el cual se cifra toda la base de datos. Además, tal y como se detallará en el apartado ‘4.1 Módulo Inicio’, la contraseña deberá cumplir unos requisitos mínimos de seguridad, asegurando así que la clave derivada generada sea robusta, y que por tanto, sea muy difícil que un agente externo rompa la contraseña mediante un ataque de fuerza bruta.

El cifrado de clave simétrica suele usar cifrado de bloques. Un cifrado de bloques es una función de cifrado simétrica que se aplica a bloques. Dentro del cifrado de bloques, se ha decidido emplear el modo CBC (*Cipher-Block Chaining*) [CBC]. Se trata de un método que emplea el resultado del bloque anterior como vector para cifrar el actual.

Como todos los cifrados de bloques, emplea un tamaño de bloque fijo, por lo que se ha utilizado *padding PKCS5* [PKCS5] para que rellenen con valores pseudoaleatorios los bytes correspondientes hasta que el último bloque complete su tamaño.

Con todos los mecanismos que se acaban de describir, se considera que la base de datos queda protegida ante ataques externos, y que por tanto, se protege de forma segura la información sensible que el usuario comparte con la aplicación al realizar alguna de las funcionalidades que esta permite.



# Capítulo 4

## Descripción detallada de los módulos del sistema

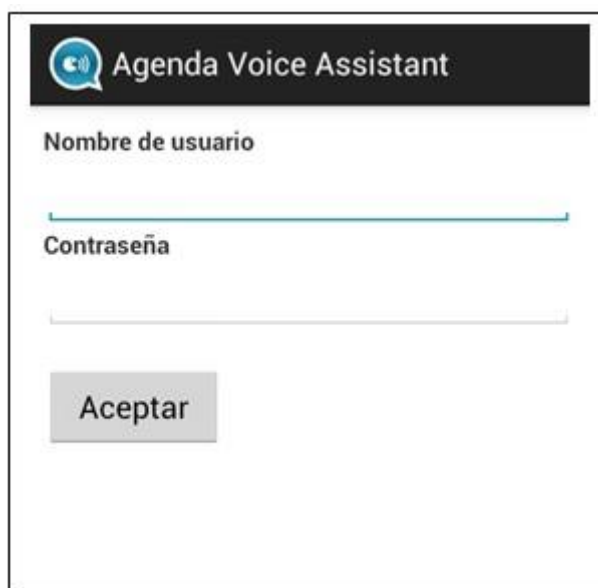
En este capítulo se analizan en detalle cada uno de los módulos de la aplicación desarrollada. Dentro de cada módulo se describen las funcionalidades que se proporcionan, se comentan sus arquitecturas y se exponen sus escenarios de uso.

### 4.1 Módulo Inicio

#### 4.1.1 Funcionalidad

El **módulo Inicio** hace referencia a la primera interfaz a la que el usuario accede al iniciar la aplicación. Este módulo es el encargado de registrar a un usuario o de validarlo, dependiendo de si es la primera vez o no que accede a la aplicación. Para ambos casos, la interfaz será la misma, aunque no así su funcionalidad. Este módulo es el único, junto en el módulo Configuración, en el que no se puede realizar las acciones por voz, debido a que ambos módulos tratan con la contraseña del usuario.

En la Figura 4.1 se puede visualizar la interfaz correspondiente a este módulo.



**Figura 4.1 Interfaz correspondiente al módulo Inicio**

## **4.1.2 Arquitectura**

En esta sección se especifican las acciones que se llevan a cabo para registrar al usuario o validarlo respectivamente.

### **4.1.2.1 Registro**

A continuación se enumeran las acciones que se realizan para registrar a un usuario.

- **Carga de la interfaz correspondiente al módulo Inicio**

Cuando el usuario quiera acceder a la aplicación por primera vez, se le pedirá que introduzca un nombre con el que la aplicación se referirá a él a partir de ahora y una contraseña que será utilizada para cifrar la información sensible de la base de datos. Una vez el usuario haya rellenado los campos con la información correspondiente pulsará el botón ‘Aceptar’.

- **Comprobación de los datos introducidos**

A continuación la aplicación comprueba que los campos cumplen unos requisitos mínimos de seguridad, con el fin de otorgarle robustez en lo que se refiere a la seguridad de la aplicación. El único requisito que debe cumplir el pseudónimo introducido por el usuario es que contenga al menos tres caracteres.

Sin embargo, las restricciones impuestas a la contraseña que elija el usuario son superiores, ya que una contraseña débil propiciaría que cualquier agente externo pudiera acceder romper el cifrado con cierta facilidad y acceder así a la información que se almacena en la base de datos. A continuación se enumeran las restricciones que se han considerado que debe cumplir la contraseña:

- Debe tener al menos 6 caracteres.
- Debe tener al menos una letra minúscula y otra mayúscula.
- Debe tener al menos un número.
- Debe tener al menos un signo de puntuación, entendiendo como signo de puntuación el punto ‘.’ y el punto y coma ‘;’.

Si alguno de estos requisitos no se cumple, se mostrará un mensaje fijo en la pantalla, indicándole al usuario el motivo por el que el nombre y/o contraseña no se han considerado válidos. Además, esta alerta también se realizará mediante voz, ya que por defecto la aplicación se ejecuta por primera vez con la opción de interactuar con el usuario habilitada.

En el caso de que los datos sean correctos, se producirá un saludo de bienvenida personalizado por parte de la aplicación. Después de este saludo, el usuario estará en el módulo Contactos.

#### **4.1.2.2 Validación**

A continuación se citan las acciones que se llevan a cabo para validar a un usuario ya registrado en la aplicación.

- **Carga de la interfaz correspondiente al módulo Inicio**

En caso de que el usuario quiera acceder a la aplicación pero ya se haya registrado previamente; es decir, no sea la primera vez que ejecuta el programa desarrollado, se le pedirá que rellene los campos con los mismos datos con los que se registró. Una vez introducidos el usuario pulsará el botón ‘Login’.

- **Comprobación de los datos introducidos**

A continuación se cotejan con los almacenados en la base de datos. Esta comprobación se realizará con los datos en claro, es decir, en lugar de cifrar los datos introducidos por el usuario, se descifrarán los almacenados en la base de datos. De esta manera se elimina cualquier posibilidad de que se obtenga el mismo *hash* para dos palabras distintas.

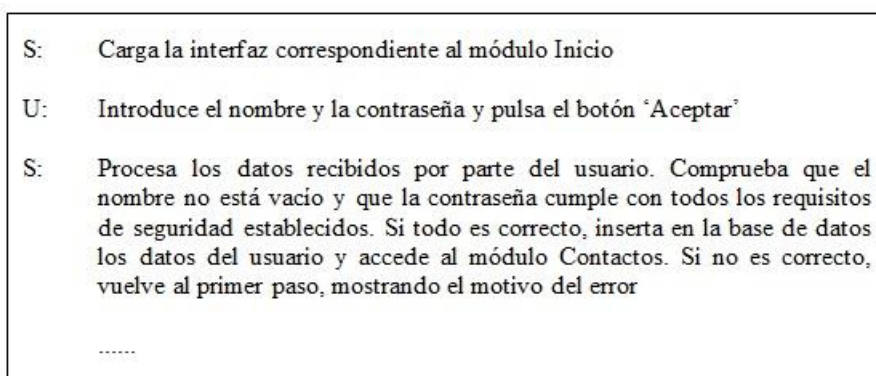
Si el resultado de la comprobación no es satisfactorio, los datos introducidos no son iguales que con los almacenados en la base de datos, se le mostrará un mensaje fijo en la pantalla indicándole al usuario la resolución de la comprobación. Independientemente de si el dato erróneo era el nombre o la contraseña el mensaje de

error será el mismo. De esta manera no se le darán pistas a un posible atacante sobre qué dato introdujo correctamente y cual no.

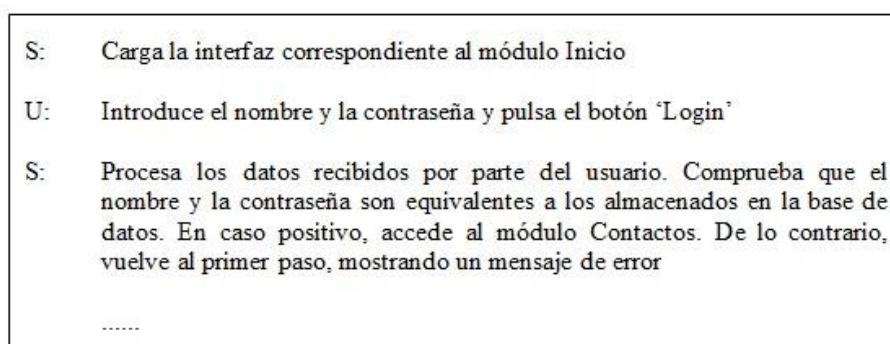
Al igual que ocurría para el registro, si la comprobación es satisfactoria se producirá un saludo personalizado por parte de la aplicación. Dependiendo de la sección del día en la que se produzca la validación la aplicación le dará los buenos días, las buenas tardes o las buenas noches. Después de dicho saludo, el usuario estará en el módulo Contactos.

### 4.1.3 Escenarios de uso

En la Figura 4.2 y Figura 4.3 se muestra cómo se realiza la comunicación entre la aplicación y el usuario para registrar o validar a un usuario respectivamente. La letra ‘S’ indica las acciones realizadas por el sistema, mientras que la letra ‘U’ muestra las acciones realizadas por el usuario.



**Figura 4.2 Escenarios de uso del módulo Inicio. Registro de usuario**



**Figura 4.3 Escenarios de uso del módulo Inicio. Validación de usuario**



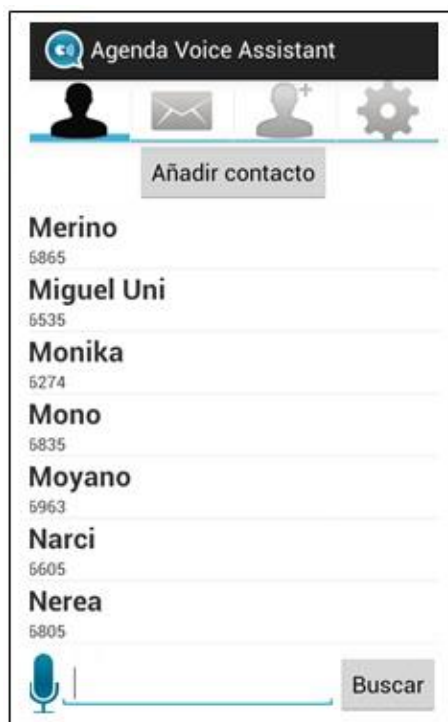
## 4.2 Módulo Contactos

### 4.2.1 Funcionalidad

Se trata del módulo central de la aplicación desarrollada. Desde él se puede acceder a todas las funcionalidades que ofrece la aplicación de forma directa o indirecta. Tal y como se ha comentado anteriormente, una vez registrado o validado correctamente el usuario en la aplicación, se le remite a la interfaz de Contactos. Esta interfaz forma parte de uno de los cuatro módulos principales: **Contactos**, **Bandeja Mensajes**, **Frecuentes** y **Configuración**.

En la pantalla de Contactos se cargan automáticamente los contactos de la agenda del usuario, mostrándole una lista ordenada alfabéticamente según el nombre e indicando también el número de teléfono. Además de la lista de contactos, también se muestra arriba un botón que da la posibilidad de añadir un nuevo contacto a la agenda.

En la Figura 4.4 se muestra la interfaz correspondiente al módulo Contactos



**Figura 4.4** Interfaz correspondiente al módulo Contactos

Como se puede observar en la imagen anterior, en la parte inferior de esta pantalla se muestran dos botones y un campo a rellenar en medio. El botón de la izquierda, con la imagen de un micrófono, sirve para activar el reconocedor de voz. El botón de la derecha sirve para activar la búsqueda de un contacto según el texto introducido en el campo central.

Pulsando el botón del micrófono, la aplicación inicia una conversación con el usuario preguntándole sobre qué acción desea realizar. El abanico de posibilidades que tiene el usuario es bastante amplio, ya que puede elegir entre añadir un nuevo contacto, editar, borrar o llamar a un contacto existente, abrir una ventana con las conversaciones que se han mantenido con un contacto en concreto, y poder acceder al resto de módulos principales anteriormente citados.

## 4.2.2 Arquitectura

A continuación se especifican las acciones que se llevan a cabo una vez el usuario haya pulsado sobre el botón con la imagen del micrófono.

- **Inicio de conversación**

La aplicación le ofrece su colaboración al usuario para ayudarle a realizar una acción determinada. Si está habilitada la opción de que la aplicación interactúe con el usuario este ofrecimiento se realizará mediante la voz, preguntándole al usuario qué acción desea realizar.

A continuación, se iniciará el reconocedor de voz de Google, para que el usuario indique mediante su voz la acción deseada. Como se comentó en la sección 3.3.1, este servicio estará funcionando siempre que el usuario esté hablando. Cuando deje de comunicarse con la aplicación, la aplicación procesará la entrada recibida.

- **Especificación de la acción a realizar**

Para realizar cualquier acción, el usuario deberá decir únicamente el nombre de la acción que desea realizar.

En caso de que la acción que requiera el nombre de un contacto (editar, borrar, llamar o enviar mensaje), el resultado de la búsqueda del contacto tiene que ser única; es decir, no debe haber más contactos que contengan ese mismo nombre.

En caso de que el resultado de la búsqueda no sea un único contacto, se le indicará al usuario que repita el nombre del contacto deseado, detallando, si es posible, el nombre para acotar mejor la búsqueda. Si aun así el resultado no fuese único, se posibilita al usuario pulsar sobre el nombre del contacto deseado. Realizando esta acción accederá al módulo que indico mediante la voz (editar, borrar, llamar o enviar mensaje).

- **Ejecución de la acción demandada**

Una vez procesada la entrada de audio emitida por el usuario, la aplicación ejecutará la acción demandada. Como se ha especificado anteriormente, para aquellas opciones que sea necesario indicar el nombre del contacto sobre el que se desea realizar una acción determinada, el resultado de la búsqueda tiene que ser único, ya que sino la aplicación no sabría a cuáles de las coincidencias se refiere el usuario. En caso de que no sea único, se volverá al primer paso, en donde la aplicación le pedirá al usuario que vuelva a especificar qué acción desea realizar.

Dependiendo de la acción demanda, la aplicación accederá a uno de los siguientes módulos: **Editar Contacto**, **Borrar Contacto**, **Añadir Contacto**, **Llamar**, **Enviar Mensaje**, **Bandeja Mensajes**, **Frecuentes** o **Configuración**.

### 4.2.3 Escenarios de uso

A continuación se muestran las distintas acciones que se pueden llevar a cabo en el módulo Contactos. Para simplificar dichos escenarios, se agruparán en dos tipos, uno en el que se realizará una acción en la que es necesario introducir el nombre del contacto, y otro en el que no es necesario el tipo del contacto.

En la Figura 4.5 se muestra un ejemplo de las acciones que deben realizar la aplicación y el contacto si se quiere editar un contacto existente en la agenda del usuario mediante mandatos de voz.

S:	Carga la interfaz correspondiente al módulo Contactos
U:	Pulsa el botón del micrófono
S:	La aplicación inicia el reconocedor de voz de Google. En caso de que esté habilitada la opción de interactuar con el usuario, la aplicación le pide al usuario, mediante un mandato de voz, que le indique la acción que desea realizar
U:	El usuario dice 'editar' y el nombre del contacto que quiere editar
S:	La aplicación busca en la agenda el nombre del contacto. En caso de que la búsqueda no dé un único resultado, se vuelve al paso 3. En caso afirmativo se accede al módulo Editar Contacto
	.....

**Figura 4.5 Escenarios de uso del módulo Contactos. Editar Contacto**

En la Figura 4.6 se muestran las acciones que deben realizar tanto el sistema como el usuario de la aplicación para añadir un nuevo contacto a la agenda telefónica.

S:	Carga la interfaz correspondiente al módulo Contactos
U:	Pulsa el botón del micrófono
S:	La aplicación inicia el reconocedor de voz de Google. En caso de que esté habilitada la opción de interactuar con el usuario, la aplicación le pide al usuario, mediante un mandato de voz, que le indique la acción que desea realizar.
U:	El usuario dice 'añadir contacto'
S:	La aplicación accede al módulo Añadir Contacto
.....	

**Figura 4.6 Escenarios de uso del módulo Contactos. Añadir Contacto**

## 4.3 Módulo Bandeja Mensajes

### 4.3.1 Funcionalidad

En este módulo se cargan automáticamente todos los mensajes que el usuario ha enviado con la aplicación y todos los recibidos desde que se la descargó. Los mensajes se muestran en una lista ordenada según la fecha de envío o recibo, es decir, el último mensaje enviado o recibido se mostrará el primero en la lista. Además, en caso de que existan varios mensajes con un mismo usuario, únicamente se indicará el último.

Desde este módulo, el usuario podrá acceder a cualquiera de los otros tres módulos principales: **módulo Inicio**, **módulo Frecuentes** y **módulo Configuración**.

La Figura 4.7 muestra la interfaz correspondiente al módulo Bandeja Mensajes.



**Figura 4.7 Interfaz correspondiente al módulo Bandeja Mensajes**

Como se puede observar en la imagen anterior, la parte inferior de la pantalla tiene el mismo formato que su homónima del módulo Contactos. Sin embargo, las opciones que se ofrecen pulsando el botón del micrófono son más reducidas, debido a que el usuario únicamente podrá realizar acciones relacionadas con el envío de mensajes.

Si el usuario decidiera buscar un contacto, el procedimiento sería el mismo que el explicado en el módulo anterior en el caso de que no se encontrase una única coincidencia. Además, la búsqueda para este módulo se realizaría únicamente entre los contactos con los que el usuario se ha comunicado mediante mensajes de texto, no con todos los contactos de su agenda.

Como se ha comentado anteriormente, en este módulo sólo se puede enviar mensajes a contactos que aparezcan en la bandeja de mensajes o bien seleccionar la opción 'Nuevo Mensaje' para enviar un mensaje introduciendo un número de teléfono, lo que le permitirá enviar un mensaje de texto a cualquier otro usuario, lo tenga o no en su agenda telefónica. En caso de que el número telefónico introducido no se corresponda con ningún contacto existente en la agenda del usuario, se creará un nuevo contacto con dicho número telefónico y nombre 'Desconocido'.

Además, cada vez que se accede a este módulo, se comprobará si los contactos con los que se ha comunicado el usuario siguen en su agenda telefónica o no. En caso afirmativo se actualizará el nombre del contacto en caso que sea necesario. En caso negativo, se eliminará el contacto de la bandeja de mensajes, pero no así de la base de datos, permitiendo así que si en un futuro el usuario vuelve a añadir al contacto en su agenda telefónica tenga cargados los mensajes que se había enviado con anterioridad.

## **4.3.2 Arquitectura**

En este apartado se especifican las acciones que se llevan a cabo una vez el usuario haya pulsado el botón del micrófono. Para poder observar mejor la diferencia entre enviar un mensaje a un contacto existente en la bandeja de mensajes y enviar un mensaje introduciendo el número telefónico, se mostrará la arquitectura de cada uno de los dos procedimientos por separado.

### **4.3.2.1 Enviar mensaje a un contacto existente de la bandeja**

- **Inicio de conversación**

La aplicación le ofrece su colaboración al usuario para ayudarle a realizar una acción determinada. Si está habilitada la opción de que la aplicación interactúe con el usuario este ofrecimiento se realizará mediante la voz, preguntándole al usuario qué acción desea realizar. A continuación se inicia el reconocedor de voz de Google.

- **Especificación de la acción a realizar**

Una se haya lanzado el servicio, si el usuario desea enviar un mensaje a un contacto de su bandeja, deberá decir su nombre. En caso de que quiera acceder a otro módulo principal de la aplicación (módulo Contactos, módulo Frecuentes o módulo Configuración) deberá realizar los mismos pasos especificados en el módulo anterior.

- **Ejecución de la acción demandada**

Una vez procesada la entrada de audio, la aplicación procederá a realizar la acción demandada. En caso de que el usuario no haya seguido el procedimiento habitual para acceder a cualquiera de los otros módulos principales, la aplicación procederá a buscar si hay algún contacto coincidente con la entrada procesada. Si se encuentra más de una coincidencia se volverá al primer paso, en el que la acción le pedirá al usuario que por favor le indique qué acción desea realizar. Si nuevamente se siguen encontrando dos o más contactos coincidentes, la aplicación aconsejará al usuario que introduzca el nombre por teclado, o bien que pulse sobre el nombre del contacto al que desea enviar un mensaje. Tanto si se pulsa el nombre del contacto, como si la búsqueda da un único contacto coincidente, la aplicación accederá al módulo Enviar Mensaje.

### 4.3.2.2 Enviar mensaje introduciendo número telefónico

- **Inicio de conversación**

Si está la opción de interactuar con el usuario, la aplicación emitirá un mensaje de voz preguntándole al usuario qué acción desea realizar. A continuación, tanto si esta opción está habilitada como si no, se iniciará el reconocedor de voz de Google.

- **Especificación de la acción a realizar**

Para acceder a la posibilidad de enviar un mensaje a un contacto, introduciendo su número de teléfono, el usuario únicamente deberá decir ‘Nuevo mensaje’. Con este simple mandato la aplicación ya sabrá cómo debe actuar.

- **Ejecución de la acción demandada**

Una vez procesada la entrada de audio, si se realiza la acción que se acaba de especificar, la aplicación accederá a una nueva interfaz, en la que se aparecerá un único campo con el nombre de ‘Número de teléfono’.

- **Especificación del número telefónico**

En esta interfaz se repetirá un procedimiento similar al anterior. El usuario deberá pulsar el botón del micrófono para interactuar con la aplicación. Una vez el servicio de reconocimiento de voz de Google vuelva a estar activo el usuario dirá el número de teléfono al que desea enviar un mensaje.

- **Procesamiento de los datos**

A continuación la aplicación verificará que los datos introducidos en ese campo se corresponden con un número. En caso afirmativo se accederá al módulo Enviar Mensaje. En caso negativo se le pedirá al usuario que repita el número telefónico. Si se siguiera produciendo una mala verificación de los datos, la aplicación aconsejaría al usuario que introdujese los datos manualmente.

### 4.3.3 Escenarios de uso

En este apartado se muestra cómo serían las conversaciones para realizar las dos acciones anteriores respectivamente. En la Figura 4.8 se observa el escenario de uso para la acción de enviar un mensaje de texto para un contacto que se encuentra en la bandeja de mensajes.

S:	Carga la interfaz correspondiente al módulo Bandeja Mensajes
U:	Pulsa el botón del micrófono
S:	La aplicación inicia el reconocedor de voz de Google. En caso de que esté habilitada la opción de interactuar con el usuario, la aplicación le pide al usuario, mediante un mandato de voz, que le indique la acción que desea realizar
U:	El usuario dice el nombre del contacto
S:	La aplicación procesa la información recogida. Si existe una única coincidencia, la aplicación accede al módulo Enviar Mensaje. En caso contrario se vuelve al paso 3
.....	

**Figura 4.8 Escenarios de uso del módulo Bandeja Mensaje. Enviar mensaje a un contacto existente**

En la Figura 4.9 se muestra las acciones que se debe realizar para enviar un mensaje previa introducción del número telefónico.

S:	Carga la interfaz correspondiente al módulo Bandeja Mensajes
U:	Pulsa el botón del micrófono
S:	La aplicación inicia el reconocedor de voz de Google. En caso de que esté habilitada la opción de interactuar con el usuario, la aplicación le pide al usuario, mediante un mandato de voz, que le indique la acción que desea realizar
U:	El usuario dice 'nuevo mensaje'
S:	La aplicación carga una nueva interfaz, con un único campo 'Número de teléfono'
U:	Pulsa el botón del micrófono
S:	La aplicación inicia el reconocedor de voz de Google
U:	El usuario dice el número de teléfono al que desea enviar el mensaje
S:	La aplicación procesa la información recibida. En caso de que los datos correspondan a un número real, se accede al módulo Enviar Mensaje. En caso contrario se vuelve al paso 7
.....	

**Figura 4.9 Escenarios de uso del módulo Bandeja Mensaje. Enviar mensaje introduciendo el número telefónico**



## 4.4 Módulo Frecuentes

### 4.4.1 Funcionalidad

Este módulo tiene una funcionalidad similar a la del módulo Contactos, ya que se pueden realizar las mismas acciones a excepción de la correspondiente al módulo Añadir Contacto. La principal diferencia de ambos módulos radica en la información que muestra por pantalla, ya que a diferencia del módulo Contactos, en este módulo no se carga la agenda de contactos del usuario, sino que se accede a la base de datos para obtener una lista de los contactos con los que el usuario se ha comunicado.

Esta lista estará ordenada por número de comunicaciones; es decir, el contacto con el que más se haya comunicado el usuario estará en primer lugar. Hay que destacar, que para obtener este número de comunicaciones, se contabilizan las veces que el usuario envía un mensaje o realiza una llamada. Por lo tanto, un contacto que se comunique en demasía con el usuario, pero esta comunicación no sea recíproca, no aparecerá en la pantalla del módulo Frecuentes.

En la Figura 4.10 se muestra la interfaz correspondiente al módulo Frecuentes.



Figura 4.10 Interfaz correspondiente al módulo Frecuentes

## 4.4.2 Arquitectura

Debido a que las funcionalidades de ambos módulos son bastante similares, la arquitectura de este módulo será equivalente a la que se explicó en el módulo Contactos.

### ● Inicio de conversación

La aplicación le ofrece su colaboración al usuario para ayudarlo a realizar una acción determinada. Si está habilitada la opción de que la aplicación interactúe con el usuario este ofrecimiento se realizará mediante la voz, preguntándole al usuario qué acción desea realizar.

A continuación, se iniciará el reconocedor de voz de Google, para que el usuario indique mediante su voz la acción deseada. Como se comentó en la sección 3.3.1, este servicio estará funcionando siempre que el usuario esté hablando. Cuando deje de comunicarse con la aplicación, la aplicación procesará la entrada recibida.

### ● Especificación de la acción a realizar

Para realizar cualquier acción, el usuario deberá decir únicamente el nombre de la acción que desea realizar.

En caso de que la acción que requiera el nombre de un contacto (editar, borrar, llamar o enviar mensaje), el resultado de la búsqueda del contacto tiene que ser única, es decir, no debe haber más contactos que contengan ese mismo nombre.

Si por el contrario de la búsqueda no se obtiene un único contacto, se le indicará al usuario que repita el nombre del contacto deseado, detallando, si es posible, el nombre para acotar mejor la búsqueda. Si aun así el resultado no fuese único, se le da la posibilidad al usuario de pulsar sobre el nombre del contacto deseado. Realizando esta acción accederá al módulo que indicó mediante la voz (editar, borrar, llamar o enviar mensaje).

### ● Ejecución de la acción demandada

Una vez procesada la entrada de audio emitida por el usuario, la aplicación ejecutará la acción demandada. Como se ha especificado anteriormente, para aquellas opciones que sea necesario indicar el nombre del contacto sobre el que se desea realizar una acción determinada, el resultado de la búsqueda tiene que ser único, ya que sino la aplicación no sabría a cuáles de las coincidencias se refiere el usuario. En caso de que no sea único, se volverá al primer paso, en donde la aplicación le pedirá al usuario que vuelva a especificar qué acción desea realizar.

Dependiendo de la acción demandada, la aplicación accederá a uno de los siguientes módulos: **Editar Contacto**, **Borrar Contacto**, **Llamar**, **Enviar Mensaje**, **Bandeja Mensajes**, **Frecuentes** o **Configuración**.

### 4.4.3 Escenarios de uso

En la Figura 4.11 se muestra un ejemplo de la comunicación existente entre la aplicación y el usuario si éste desea realizar una llamada telefónica a un contacto determinado.

S:	Carga la interfaz correspondiente al módulo Frecuentes
U:	Pulsa el botón del micrófono
S:	La aplicación inicia el reconocedor de voz de Google. En caso de que esté habilitada la opción de interactuar con el usuario, la aplicación le pide al usuario, mediante un mandato de voz, que le indique la acción que desea realizar
U:	El usuario dice 'llamar a' y el nombre del contacto al que desea llamar
S:	La aplicación busca contactos coincidentes con la entrada recibida en la lista de contactos frecuentes. Si encuentra más de un contacto, se volverá al paso 3. En caso satisfactorio se accede al módulo Llamar
	.....

Figura 4.11 Escenario de uso del módulo Frecuentes. Llamar a un contacto

## 4.5 Módulo Configuración

### 4.5.1 Funcionalidad

Desde este módulo el usuario podrá realizar cambios en la configuración de la aplicación desarrollada. Estos cambios se basan en la habilitación o no de la opción de que la aplicación interactúe con el usuario y en el cambio de contraseña.

En la Figura 4.12 se muestra la interfaz principal del módulo Configuración. En ella se puede apreciar la existencia de un nuevo botón de micrófono. Pulsando este botón el usuario podrá realizar los dos cambios en la configuración anteriormente mencionados mediante mandatos de voz.



**Figura 4.12 Interfaz principal del módulo Configuración**

Como se ha comentado en módulos anteriores, la opción de interactuar con la aplicación es clave para que la aplicación se comuniquen con el usuario de forma oral. Si esta opción está deshabilitada la aplicación no emitirá mensajes de voz. Esto se consigue gracias a un campo de la base de datos llamado 'audio'.

Todos los módulos acceden a dicho campo para saber si está habilitada o no la opción. Sin embargo, únicamente es posible cambiar el estado de ese campo desde este módulo.

Por otra parte, la aplicación también brinda la oportunidad al usuario de cambiar la contraseña con la que se registró. De esta manera, el usuario podrá cambiar la contraseña tantas veces como quiera, haciendo casi imposible para un atacante externo el robo de la misma.

En la Figura 4.13 se muestra la interfaz correspondiente al cambio de contraseña del módulo Configuración. Como se aprecia, es necesario introducir la contraseña antigua así como la nueva de forma repetida. Para esta nueva contraseña se volverán a requerir los requisitos mínimos de seguridad que se especificaron en el módulo Inicio, es decir, la contraseña deberá tener una longitud superior a cinco caracteres, y deberá contener al menos una letra mayúscula, una letra minúscula, un signo de puntuación y un número.

**Figura 4.13 Interfaz correspondiente al cambio de contraseña del módulo Configuración**

Nuevamente, toda esta fase de cambio de contraseña se realiza sin dar la opción al usuario de ejecutar la acción mediante un mandato de voz, ya que se vería afectada la seguridad. Si todo este procedimiento se realiza de forma satisfactoria, se procederá al cifrado de la base de datos con esta nueva contraseña, descifrándola previamente con la contraseña antigua.

## **4.5.2 Arquitectura**

A continuación se enumeran las acciones que se han de llevar a cabo para realizar la habilitación/deshabilitación de la opción de audio y el cambio de contraseña respectivamente.

### **4.5.2.1 Habilitación de la opción de audio**

Las principales acciones que se llevan a cabo para habilitar el audio en la aplicación se especifican a continuación.

- **Inicio de conversación**

La aplicación le ofrece su colaboración al usuario para ayudarlo a realizar una acción determinada. Si está habilitada la opción de que la aplicación interactúe con el usuario este ofrecimiento se realizará mediante la voz, preguntándole al usuario qué acción desea realizar.

A continuación, se iniciará el reconocedor de voz de Google, para que el usuario indique mediante su voz la acción deseada. Como se comentó en la sección 3.3.1, este servicio estará funcionando siempre que el usuario esté hablando. Cuando deje de comunicarse con la aplicación, la aplicación procesará la entrada recibida.

- **Especificación de la acción a realizar**

Una vez iniciado el servicio de reconocimiento de voz el usuario el usuario realizará una petición oral acerca de la acción que desea realizar. Para cambiar el estado de la opción de que la aplicación interactúe con el usuario, éste únicamente deberá decir ‘Habilitar’ o ‘Deshabilitar’, dependiendo del estado que quiere que adquiera esta opción.

- **Ejecución de la acción demandada**

Una vez procesada la entrada de audio emitida por el usuario, la aplicación ejecutará la acción demandada. En caso de que la opción sea habilitar el audio y el audio ya estuviera habilitado previamente, la aplicación le advertirá al usuario de forma oral de este suceso.

#### **4.5.2.2 Cambiar contraseña**

A continuación se enumera todo el procedimiento que se ha de llevar a cabo para realizar el cambio de contraseña en la aplicación.

- **Inicio de conversación**

Si está la opción de interactuar con el usuario, la aplicación emitirá un mensaje de voz preguntándole al usuario qué acción desea realizar. A continuación, tanto si esta opción está habilitada como si no, se iniciará el reconocedor de voz de Google.

- **Especificación de la acción a realizar**

Para acceder a la posibilidad de enviar un mensaje a un contacto, introduciendo su número de teléfono, el usuario únicamente deberá decir ‘Cambiar contraseña’. Con este simple mandato la aplicación ya sabrá cómo debe actuar.

- **Ejecución de la acción demandada**

Una vez procesada la entrada de audio, si se realiza la acción que se acaba de especificar, la aplicación accederá a una nueva interfaz, Figura 4.13, que el usuario deberá rellenar.

- **Inserción de los datos de forma manual**

Como se ha comentado anteriormente esta interfaz no posibilita la opción de realizar el mandato mediante la voz, ya que se considera que se podría ver en peligro la seguridad de la aplicación, pues en la contraseña radica toda la fuerza del cifrado de la base de datos. Por lo tanto, el usuario deberá rellenar los campos con los datos correspondientes.

- **Procesamiento de los datos**

A continuación la aplicación verificará que los dos campos habilitados para la nueva contraseña son idénticos, y que la antigua contraseña que se introduce es equivalente a la guardada en la base de datos. Si todo esto es correcto, se cifra la base de datos con la nueva contraseña, actualizándose el valor de la misma. A continuación la aplicación accederá a la interfaz anterior, Figura 4.12.

### 4.5.3 Escenarios de uso

En la Figura 4.14 y Figura 4.15 se muestran los escenarios de uso para las acciones de habilitar la opción de audio y cambiar la contraseña respectivamente.

S:	Carga la interfaz correspondiente al módulo Configuración
U:	Pulsa el botón del micrófono
S:	La aplicación inicia el reconocedor de voz de Google. En caso de que esté habilitada la opción de interactuar con el usuario, la aplicación le pide al usuario, mediante un mandato de voz, que le indique la acción que desea realizar
U:	El usuario dice 'habilitar audio'
S:	La aplicación verifica que el audio está deshabilitado y cambia su estado. En caso de que el audio ya estuviese habilitado se le notifica al usuario mediante un mensaje de voz
	.....

**Figura 4.14 Escenarios de uso del módulo Configuración. Habilitar audio**

S:	Carga la interfaz correspondiente al módulo Configuración
U:	Pulsa el botón del micrófono
S:	La aplicación inicia el reconocedor de voz de Google. En caso de que esté habilitada la opción de interactuar con el usuario, la aplicación le pide al usuario, mediante un mandato de voz, que le indique la acción que desea realizar
U:	El usuario dice 'cambiar contraseña'
S:	La aplicación carga la nueva interfaz, en donde se muestra una serie de campos a rellenar correctamente por el usuario
U:	El usuario rellena los campos con la contraseña antigua y la nueva contraseña. Todo esto lo realiza de forma manual, mediante el teclado virtual del dispositivo móvil
S:	La aplicación verifica que todo está correcto. En caso afirmativo se vuelve al paso 1. De lo contrario, se vuelve al paso 5 indicando en un mensaje el motivo del error
	.....

**Figura 4.15 Escenarios de uso del módulo Configuración. Cambiar contraseña**

## 4.6 Módulo Llamar

### 4.6.1 Funcionalidad

Este módulo es el encargado de implementar la funcionalidad de llamar. Se trata del único junto con el módulo Borrar en los que se requiere una previa colaboración del usuario y no tienen una interfaz propia.

Únicamente es posible acceder a esta funcionalidad desde los módulos Contactos y Frecuentes. Por lo tanto, cuando se accede a esta función ya está seleccionado el contacto al que se va a llamar.

### 4.6.2 Arquitectura

A continuación se comenta la arquitectura de este módulo. Cabe destacar que todos los pasos previos antes de la acción de llamar a un contacto se llevan a cabo en el módulo Contactos o en el módulo Frecuentes. Por lo tanto este módulo únicamente realiza la acción de llamar.



- **Recogida de los datos de contacto y acción de llamar**

Como se ha comentado anteriormente, para acceder al módulo previamente hay que especificar o en el módulo Contactos o en el módulo Frecuentes el nombre del contacto al que se quiera llamar. Debido a ese motivo, en este módulo no se requiere más colaboración por parte del usuario que la obtenida en el módulo antecesor.

### 4.6.3 Escenarios de uso

Para que sea visualmente más fácil comprender todo el proceso necesario para realizar una llamada, en la Figura 4.16 se parte de la situación de que el usuario se encuentra en el módulo Contactos, ya con el botón del micrófono pulsado.

U:	El usuario se encuentra en el módulo Contactos y pulsa el botón del micrófono
S:	La aplicación inicia el reconocedor de voz de Google. En caso de que esté activa la opción de interactuar con el usuario, la aplicación pregunta al usuario, mediante un mensaje de voz, qué acción desea realizar
U:	El usuario dice 'Llamar' y el nombre del contacto al que desea llamar
S:	La aplicación accede al módulo Llamar, pasándole el número telefónico correspondiente a dicho contacto
S:	Una vez en el módulo Llamar, la aplicación ejecuta la acción de llamar y accede al menú Contactos
.....	

**Figura 4.16 Escenario de uso del módulo Llamar**

## 4.7 Módulo Añadir Contacto

### 4.7.1 Funcionalidad

En este módulo se realiza toda la gestión relacionada con la inserción de un nuevo contacto en la agenda telefónica del usuario de la aplicación desarrollada.

La Figura 4.17 muestra la interfaz correspondiente al módulo Añadir Contacto.



**Figura 4.17** Interfaz correspondiente al módulo Añadir Contacto

Como se puede observar en la imagen anterior, en esta pantalla aparecen dos campos que el usuario ha de rellenar. Uno con el nombre del contacto y otro con el número de teléfono.

Ambas acciones se pueden realizar mediante mandatos de voz proporcionados en acciones distintas. Es decir, para realizar el proceso de inserción del nombre del nuevo contacto el usuario deberá pulsar el botón del micrófono que aparece al lado del campo del nombre. El mismo procedimiento habría que seguir para rellenar el campo del número telefónico, pero pulsando en el botón del micrófono situado al lado de este campo.

Para que esta acción se realice satisfactoriamente, la aplicación verificará que los datos introducidos en el campo del número de teléfono son efectivamente valores numéricos. Así mismo, si el usuario decide insertar el contacto sin haber rellenado uno o los dos campos, la

aplicación le advertirá mediante un mensaje de voz, si así lo tiene estipulado el usuario, y mediante un mensaje fijo de error. En dicho mensaje se le notificará el motivo de por qué no ha sido posible insertar el contacto en la agenda.

## 4.7.2 Arquitectura

En este apartado se especifican las acciones necesarias para que el usuario pueda añadir un contacto a su agenda telefónica. Para agilizar este procedimiento, en la arquitectura se omite todo lo relacionado al inicio del servicio de reconocimiento de voz de Google, ya que se ha explicado con anterioridad y para este módulo se ejecuta tres veces.

- **Especificación del nombre del contacto**

Una vez pulsado el botón del micrófono correspondiente al nombre del contacto, el usuario dirá el pseudónimo con el cual se conocerá al nuevo contacto.

- **Procesamiento de los datos**

La aplicación procesará los datos de entrada. En caso de que no se haya realizado con éxito el reconocimiento de voz, la aplicación le pedirá al usuario que repita el nombre del contacto. Si el problema siguiera, la aplicación le aconsejaría que introdujese el nombre de forma manual.

- **Especificación del número telefónico**

Igual que para la inserción del nombre del contacto, el usuario previamente pulsará el botón del micrófono correspondiente al número telefónico. Una vez iniciado el servicio de reconocimiento de voz de Google el usuario citará el número de teléfono del nuevo contacto.

- **Procesamiento de los datos**

Una vez insertado los datos para el número telefónico, la aplicación verificará que efectivamente se trata de un número. En caso negativo, reseteará el valor que el usuario introdujo en el campo y le pedirá al usuario que repita el número y si el problema continuase le aconsejaría, igual que ocurría en la inserción del nombre, que introdujese el número de teléfono de forma manual.

- **Especificación de la acción a realizar**

A continuación el usuario pulsará el botón del micrófono central, que realizará la misma acción que el botón ‘Añadir contacto’. Una vez se inicie el servicio de

reconocimiento de voz, el usuario especificará la opción a realizar. En este caso, la opción sería ‘Añadir contacto’.

- **Ejecución de la acción especificada**

La aplicación procesará la acción demandada por parte del usuario. Si el usuario solicita la inserción del contacto, la aplicación comprobará que ambos campos, tanto el nombre como el número de teléfono están rellenos. En caso afirmativo la aplicación realizará la acción en cuestión y accederá al módulo Contactos.

### 4.7.3 Escenarios de uso

En la Figura 4.18 se muestra el escenario de uso típico para la incorporación de un nuevo contacto a la agenda del usuario.

S:	La aplicación carga la interfaz correspondiente al módulo Añadir Contacto
U:	El usuario pulsa el botón del micrófono correspondiente al nombre del contacto. Una vez iniciado el servicio de reconocimiento de voz indica el pseudónimo con el cual se denominará al contacto
S:	La aplicación procesa los datos. Si no ha funcionado correctamente el reconocedor de voz le solicita al usuario que repita la acción anterior. En caso de que se vuelva a producir, le aconseja que lo introduzca de forma manual
U:	El usuario pulsa el botón del micrófono correspondiente al número de teléfono. Una vez iniciado el servicio de reconocimiento de voz indica el número de teléfono del contacto en cuestión
S:	La aplicación procesa los datos de entrada, para verificar que se trata de un número. En caso negativo se vuelve al paso 3, indicándole al usuario el motivo, mediante un mensaje de voz
U:	El usuario pulsa el botón central de la pantalla y solicita realizar la acción ‘añadir contacto’
S:	La aplicación procesa los datos de entrada. Si la entrada se corresponde con la acción de añadir contacto, comprueba que los campos del nombre y del número telefónico no están vacíos. En caso de que todo esté correcto la aplicación realiza la acción en cuestión y accede al módulo Contactos. En caso contrario se vuelve al paso 1 o 3 dependiendo del campo que estuviese sin rellenar
.....	

**Figura 4.18 Escenario de uso del módulo Añadir Contacto**

## 4.8 Módulo Editar Contacto

### 4.8.1 Funcionalidad

Este módulo es el encargado de implementar la funcionalidad de edición de un contacto existente en la agenda telefónica del usuario. En la Figura 4.19 se muestra la interfaz correspondiente al módulo Editar Contacto.



**Figura 4.19** Interfaz correspondiente al módulo Editar Contacto

Como se puede observar en la imagen anterior, esta pantalla es similar a la Figura 4.17, con la diferencia de que en este caso los campos relativos al nombre y al número telefónico se generan con los valores actuales del contacto en cuestión.

Nuevamente, ambas acciones se pueden realizar mediante mandatos de voz, pero se ejecutan en acciones distintas. Es decir, para realizar el proceso de edición del nombre del nuevo contacto el usuario deberá pulsar el botón del micrófono que aparece al lado del campo del nombre. El mismo procedimiento habría que seguir para editar el campo del número telefónico, pero pulsando en el otro botón del micrófono.

Para que esta acción se realice satisfactoriamente, la aplicación verificará que los datos introducidos en el campo del número de teléfono son efectivamente valores numéricos. Así mismo, si el usuario decide editar el contacto sin haber rellenado uno o los dos campos, la aplicación le advertirá mediante un mensaje de voz, si así lo tiene estipulado el usuario, y

mediante un mensaje fijo de error. En dicho mensaje se le notificará el motivo de por qué no ha sido posible editar el contacto en cuestión.

## 4.8.2 Arquitectura

En este apartado se especifican las acciones necesarias para que el usuario pueda editar un contacto a su agenda telefónica. Para agilizar este procedimiento, en la arquitectura se omite todo lo relacionado al inicio del servicio de reconocimiento de voz de Google, ya que se ha explicado con anterioridad y para este módulo se ejecuta en numerosas ocasiones. Para que se pueda observar toda su arquitectura se asume que el usuario desea editar todos los campos del contacto.

- **Carga de la interfaz correspondiente al módulo Editar Contacto**

Cuando el usuario acceda a este módulo, aparecerá el nombre y el número de teléfono actuales del contacto, de modo que si el usuario pulsa el botón ‘Editar’ sin realizar ningún cambio, no se considere modificación alguna.

- **Especificación del nombre del contacto**

Una vez pulsado el botón del micrófono correspondiente al nombre del contacto, el usuario dirá el nuevo pseudónimo con el cual se denominará al contacto.

- **Procesamiento de los datos**

La aplicación procesará los datos de entrada. En caso de que no se haya realizado con éxito el reconocimiento de voz, la aplicación le pedirá al usuario que repita el nombre del contacto. Si el problema siguiera, la aplicación le aconsejaría que introdujese el nombre de forma manual.

- **Especificación del número telefónico**

Igual que para la inserción del nombre del contacto, el usuario previamente pulsará el botón del micrófono correspondiente al número telefónico. Una vez iniciado el servicio de reconocimiento de voz de Google el usuario citará el nuevo número de teléfono del contacto.

- **Procesamiento de los datos**

Una vez insertado los datos para el número telefónico, la aplicación verificará que efectivamente se trata de un número. En caso negativo, reseteará el valor que el usuario introdujo en el campo y le pedirá al usuario que repita el número. Si el problema

persistiese, le aconsejaría, igual que ocurría en la inserción del nombre, que introdujese el número de teléfono de forma manual.

- **Especificación de la acción a realizar**

A continuación el usuario pulsará el botón del micrófono central, que realizará la misma acción que el botón ‘Editar’. Una vez se inicie el servicio de reconocimiento de voz, el usuario especificará la opción a realizar. En este caso, la opción sería ‘Editar contacto’.

- **Ejecución de la acción especificada**

La aplicación procesará la acción demandada por parte del usuario. Si el usuario solicita la edición del contacto, la aplicación comprobará que ambos campos, tanto el nombre como el número de teléfono están rellenos. En caso afirmativo la aplicación realizará la acción en cuestión y accederá al módulo Contactos.

### 4.8.3 Escenarios de uso

En la Figura 4.20 se muestra el escenario de uso típico para la adición de un nuevo contacto a la agenda del usuario.

S:	La aplicación carga la interfaz correspondiente al módulo Editar Contacto, con los valores actuales del contacto seleccionado
U:	El usuario pulsa el botón del micrófono correspondiente al nombre del contacto. Una vez iniciado el servicio de reconocimiento de voz indica el nuevo pseudónimo con el cual se denominará al contacto
S:	La aplicación procesa los datos. Si no ha funcionado correctamente el reconocedor de voz le solicita al usuario que repita la acción anterior. En caso de que se vuelva a producir, le aconseja que lo introduzca de forma manual
U:	Pulsa el botón del micrófono correspondiente al número de teléfono. Una vez iniciado el servicio de reconocimiento de voz indica el nuevo número de teléfono del contacto en cuestión
S:	Procesa los datos de entrada, para verificar que se trata de un número. En caso negativo se vuelve al paso 3, indicándole el motivo al usuario mediante un mensaje de voz
U:	Pulsa el botón central de la pantalla y solicita realizar la acción ‘Editar contacto’
S:	Procesa los datos de entrada. Si la entrada se corresponde con la acción de editar un contacto comprueba que los campos del nombre y del número telefónico no están vacíos. En caso de que todo esté correcto la aplicación realiza la acción en cuestión y accede al módulo Contactos. En caso contrario se vuelve al paso 1 o 3 dependiendo de qué campo estuviese sin rellenar
.....	

**Figura 4.20 Escenario de uso del módulo Editar Contacto**

## 4.9 Módulo Borrar Contacto

### 4.9.1 Funcionalidad

Este módulo es el encargado de implementar la funcionalidad de borrado. Como se especificó en el apartado 4.6, este módulo no tiene una interfaz propia, sino que se ejecuta de forma transparente al usuario.

Únicamente es posible acceder a esta funcionalidad desde los módulos Contactos y Frecuentes. Por lo tanto, cuando se accede a esta función ya está seleccionado el contacto que se desea borrar.

Debido a que la acción de borrado de un contacto de la agenda telefónica del usuario no tiene retorno; es decir, una vez borrado no se puede volver atrás, se le pedirá al usuario una nueva verificación de que efectivamente se desea borrar el contacto en cuestión. Si se tiene activada la opción de que la aplicación interactúe con el usuario, esta verificación se realizará también de forma sonora.

### 4.9.2 Arquitectura

A continuación se comenta la arquitectura de este módulo. Cabe destacar que todos los pasos previos antes de la acción de borrar un contacto se realizan en el módulo Contactos o en el módulo Frecuentes. Por lo tanto este módulo únicamente genera la verificación por parte del usuario para borrar el contacto en cuestión y realiza la acción de borrado.

- **Recogida de los datos del contacto y verificación de borrado**

Como se ha comentado anteriormente, para acceder al módulo previamente hay que especificar en el módulo Contactos o en el módulo Frecuentes el nombre del contacto que se quiere borrar. Debido a que la acción de borrado no tiene retorno posible, se pide al usuario una nueva confirmación de borrado. Debido a ese motivo, en este módulo no se requiere más colaboración por parte del usuario que la obtenida en el módulo antecesor.

- **Confirmación de borrado**

El usuario deberá confirmar que efectivamente desea borrar el contacto en cuestión. Dicha confirmación se llevará a cabo pulsando la opción deseada en un menú emergente. Para evitar que se borre accidentalmente un usuario, se ha desestimado la opción de que dicha verificación se pudiera realizar también de forma oral.



- **Ejecución de la acción de borrado**

Después de recibir la confirmación, la aplicación llevará a cabo la acción de borrado. Una vez realizada, accederá al módulo Contactos.

### 4.9.3 Escenarios de uso

Para que sea visualmente más fácil comprender todo el proceso necesario para borrar un contacto, en la Figura 4.21 se parte de la situación de que el usuario se encuentra en el módulo Contactos, ya con el botón del micrófono pulsado.

U:	El usuario se encuentra en el módulo Contactos y pulsa el botón del micrófono
S:	La aplicación inicia el reconocedor de voz de Google. En caso de que esté activa la opción de interactuar con el usuario, la aplicación pregunta al usuario, mediante un mensaje de voz, qué acción desea realizar
U:	El usuario dice 'Borrar' y el nombre del contacto al que desea eliminar
S:	La aplicación accede al módulo Borrar Contacto, pasándole el número telefónico correspondiente a dicho contacto
S:	Una vez en el módulo Borrar Contacto, la aplicación solicita una verificación antes de borrar el contacto en cuestión
U:	El usuario pulsa la opción de confirmar del menú emergente
S:	La aplicación ejecuta la acción de borrado y accede al menú Contactos
.....	

**Figura 4.21 Escenario de uso del módulo Borrar Contacto**

## 4.10 Módulo Enviar Mensaje

### 4.10.1 Funcionalidad

En este módulo se realiza la gestión de envío de mensajes, y la visualización de los mensajes enviados y recibidos del usuario con cada uno de los contactos.

Como se puede observar en la Figura 4.22, la interfaz de este módulo es personalizada según el contacto con el que el usuario mantiene la conversación, mostrándose el nombre del contacto en la parte superior de la pantalla. En caso de que el usuario esté guardado en la agenda como 'Desconocido', se mostrará el número de teléfono en lugar del nombre.



**Figura 4.22 Interfaz correspondiente al módulo Enviar Mensaje**

Los mensajes situados a la derecha de la imagen corresponderán a los enviados por el usuario, mientras que los situados en el margen izquierdo serán los mensajes recibidos. Además, los colores de las cajas que los contienen son distintos, lo que facilita una rápida lectura.

Así mismo, si se tiene habilitada la opción de que la aplicación interactúe con el usuario, ésta puede leernos cada mensaje de forma individualizada, indicándonos el origen (enviado o recibido), la hora y el contenido de dicho mensaje. Para ello, el usuario únicamente deberá pulsar encima del mensaje que quiera.

En lo que se refiere a la acción de envío de mensajes propiamente dicha, el usuario únicamente deberá pulsar el botón del micrófono e indicar el mensaje que se quiere enviar. Una vez realizada esta acción, la aplicación dará la opción de enviar el mensaje, editarlo, cancelarlo o de que nos lea el mensaje para corroborar el contenido del mismo.

En el caso de que se seleccione la opción de editar el mensaje, esta edición se realizará de forma manual, modificando el texto que se desee. Si se selecciona la opción de que la aplicación nos lea el mensaje, una vez terminada esta opción, nos volverá a indicar las distintas opciones que tenemos: enviar el mensaje, editarlo o cancelarlo.

Por último, existe la opción de que si el mensaje indicado termina con la frase ‘enviar mensaje’, éste se envíe automáticamente, sin que la aplicación le pida al usuario qué acción desea realizar. De esta manera, se evita un paso intermedio para aquellos usuarios más acostumbrados a la aplicación, o que requieren una mayor celeridad a la hora de enviar el mensaje de texto.

## 4.10.2 Arquitectura

A continuación se comenta la arquitectura de este módulo. Para ello se comentarán las acciones llevadas a cabo para enviar un mensaje por voz, seleccionar la opción de que la aplicación nos lea el contenido del mismo y finalmente enviarlo.

- **Inicio de la conversación**

Si está la opción de interactuar con el usuario, la aplicación emitirá un mensaje de voz pidiéndole que le indique el mensaje que le desea enviar al contacto en cuestión. A continuación, tanto si esta opción está habilitada como si no, se iniciará el reconocedor de voz de Google.

- **Especificación del contenido del mensaje**

Una vez iniciado el servicio el usuario únicamente indicará el contenido del mensaje. Si no desea enviar el mensaje de manera directa, el usuario no deberá realizar ninguna acción más en este paso.

- **Procesamiento de los datos introducidos y menú de opciones**

A continuación la aplicación habilitará un menú con las distintas opciones que se pueden realizar con el mensaje indicado anteriormente y se iniciará nuevamente el reconocedor de voz de Google.

- **Seleccionar la opción de que el mensaje sea leído**

El usuario seleccionará la opción que desee. La elección de la opción se realizará de forma oral ya que este menú únicamente aparecerá para los mensajes que se quieran enviar introduciendo el texto mediante mandatos de voz. Para que indicar que la aplicación lea el contenido del mensaje introducido el usuario dirá ‘leer’ o ‘leer mensaje’.

- **Lectura del mensaje y menú de opciones**

La aplicación procesará la respuesta dada por el usuario. En el caso de este ejemplo, una vez procesada la entrada de datos, se iniciará el servicio de conversor de texto a voz y la aplicación leerá el contenido del mensaje. Una vez haya terminado volverá a preguntar al usuario qué acción desea realizar.

- **Selección de la opción de envío del mensaje**

Nuevamente, el usuario indicará la opción que desee. Siguiendo con la especificación comentada al inicio de esta sección, en este caso el usuario seleccionará la opción de enviar el mensaje.

- **Ejecución de la acción especificada**

La aplicación procesará la nueva respuesta recibida. En el caso de que el usuario haya seleccionado la opción de enviar el mensaje, la aplicación ejecutará dicha acción. Una vez realizada, insertará en la base de datos el mensaje en cuestión y actualizará la interfaz para que aparezca el mensaje citado. Si el usuario volviese al pulsar el botón del micrófono se repetirían todos los pasos mencionados anteriormente.

### 4.10.3 Escenarios de uso

En la Figura 4.23 se muestra el proceso necesario para llevar a cabo la acción especificada en el apartado anterior.

U:	El usuario se encuentra en el módulo Enviar Mensaje y pulsa el botón del micrófono
S:	La aplicación inicia el reconocedor de voz de Google. En caso de que esté activa la opción de interactuar con el usuario, la aplicación emite un mensaje de voz pidiéndole al usuario que le indique el mensaje que desea enviar
U:	Una vez iniciado el servicio de reconocimiento de voz indica el contenido del mensaje
S:	La aplicación procesa los datos de entrada. En caso de que esté activa la opción de interactuar con el usuario, le ofrece una serie de posibilidades como son: leer, editar, borrar o enviar el mensaje. A continuación, inicia el reconocedor de voz de Google de nuevo
U:	Una vez se iniciado el servicio de reconocimiento de voz, el usuario dice 'Leer mensaje'
S:	La aplicación procesa los datos de entrada. A continuación reproduce, mediante el servicio de conversión de texto a voz, el mensaje previamente introducido. Finalizado este procedimiento, vuelve a iniciar el reconocedor de voz de Google
U:	Una vez iniciado el servicio de reconocimiento de voz, el usuario dice 'enviar mensaje'
S:	La aplicación procesa los datos de entrada y ejecuta la acción de envío del mensaje al contacto en cuestión
	.....

**Figura 4.23 Escenario de uso del módulo Enviar Mensaje**

## 4.11 Módulo Recibir Mensaje

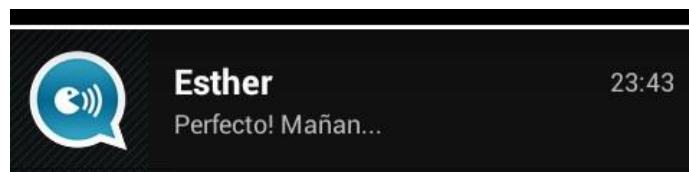
### 4.11.1 Funcionalidad

Este módulo es el único que no es una actividad propiamente dicha, sino que gestiona las funciones del Broadcast Receiver que se comentó en el apartado 3.3.3.

Cada vez que se recibe un mensaje por parte de cualquier contacto, independientemente de si el usuario lo tiene en su agenda telefónica o no, se insertará en la base de datos de la aplicación. En el caso de que el contacto no exista en la agenda del usuario, además se llamará al módulo 'Añadir Contacto' y se procederá a su inclusión en la agenda bajo el pseudónimo de 'Desconocido'. Esta acción se realizará de manera transparente al usuario; es decir, no requerirá su interacción en ningún momento.

Por último, para cada mensaje entrante se comprobará si el usuario está en ese momento manteniendo una conversación con dicho contacto. En caso de que así, si se tiene habilitada la opción de que la aplicación interactúe con el usuario, se procederá a la lectura del mensaje que se acaba de recibir. Si por el contrario el usuario no está manteniendo una conversación con el contacto en cuestión, bien porque está comunicándose con otro contacto o porque simplemente no está interactuando con la aplicación, se creará una alerta en la barra de mensaje alertando al usuario del suceso ocurrido.

Como se puede observar en la Figura 4.24, al desplegar la barra de notificaciones, aparecerá el nombre del contacto emisor del mensaje así como una fracción del texto. Si se pulsa sobre la notificación, la aplicación accederá a la interfaz correspondiente al módulo 'Enviar Mensaje' del contacto en cuestión, mostrando el mensaje recibido.



**Figura 4.24 Notificación generada al recibir un mensaje de texto**

# Capítulo 5

## Evaluación de la aplicación

En este capítulo se describe la evaluación preliminar que se ha llevado a cabo de la aplicación desarrollada para el Trabajo Fin de Grado. Para ello, se ha creado un cuestionario que recoge la valoración subjetiva de los usuarios que han hecho uso previamente de la aplicación. Los resultados obtenidos se muestran a lo largo de este capítulo en forma de gráficas.

### 5.1 Metodología de evaluación

La evaluación, el estudio del rendimiento y el análisis de la usabilidad de los sistemas de diálogo son procedimientos necesarios para minimizar costes y optimizar resultados en las aplicaciones que hagan uso de ellos.

La evaluación de la aplicación desarrollada se ha llevado a cabo a través de valoraciones de calidad. Para ello, se ha diseñado un cuestionario que recoge la opinión subjetiva y el grado de satisfacción de los usuarios tras utilizar la aplicación desarrollada, obteniendo así una evaluación cualitativa de la percepción del sistema por parte de los usuarios de la aplicación.

Con esta técnica de análisis se pretende además obtener, sin excesivo consumo de recursos, información básica acerca de la opinión de los potenciales usuarios de la aplicación. Los participantes de esta técnica de análisis han sido personas elegidas al azar con edades comprendidas entre los 35 y los 65 años, ya que se entiende que son ellos los que mejor podrían apreciar las características de la aplicación. En total, se ha requerido la colaboración de 20 usuarios potenciales. Con este número se considera cubierta todas las posibles alternativas de usuarios de la aplicación que puedan existir.

Los aspectos que se han querido analizar son: el grado en el cual valora el usuario que es entendido por el sistema y viceversa, si la aplicación es lo suficientemente intuitiva, la presencia de errores, la similitud de los mensajes de voz de la aplicación con respecto al comportamiento humano y el nivel de satisfacción con el sistema global. Además, información adicional sobre su grado de conocimiento acerca de los dispositivos móviles y uso de sistemas de diálogo, que sirven para hacerse una idea del perfil de los usuarios encuestados.

El cuestionario elaborado para este fin consta de diez preguntas (Figura 5.1). Cada pregunta tiene cinco posibles respuestas, pero sólo es posible facilitar una por cada pregunta. Antes de rellenar el cuestionario, cada usuario potencial ha hecho uso de la aplicación desarrollada de forma individual, por un tiempo aproximado de 20 minutos. Cada participante ha tenido libertad absoluta para descubrir por él mismo las distintas funcionalidades que ofrece la aplicación, sin influir sobre qué acciones debía realizar o cuales evitar. Con esto se ha querido asegurar que los encuestados iban a haber adquirido los conocimientos suficientes para poder contestar con seguridad a las preguntas que se muestran en el cuestionario, y que cada usuario iba a tener un punto de vista distinto de la aplicación, pues no todos habrían realizado exactamente las mismas acciones.

<p><b>1. Puntúe en una escala del 1 al 5 su experiencia previa usando dispositivos móviles. (1="Bajo", 5="Alto").</b></p> <p>1</p> <p>2</p> <p>3</p> <p>4</p> <p>5</p> <p><b>2. ¿Con qué frecuencia utiliza su Smartphone al cabo de un día?</b></p> <p>Nunca</p> <p>Pocas veces</p> <p>Algunas veces</p> <p>Bastantes veces</p> <p>Muchas veces</p> <p><b>3. Puntúe en una escala del 1 al 5 su experiencia previa usando Interfaces de voz o sistemas similares. (1="Bajo", 5="Alto").</b></p> <p>1</p> <p>2</p> <p>3</p> <p>4</p> <p>5</p>
---



**4. ¿Qué tal le ha entendido la aplicación?**

Muy mal

Mal

Normal

Bien

Muy bien

**5. ¿Qué tal ha entendido los mensajes generados por la aplicación?**

Muy mal

Mal

Normal

Bien

Muy bien

**6. En su opinión, la navegación por la aplicación fue...**

Muy mala

Mala

Normal

Buena

Muy buena

**7. ¿Le ha sido fácil realizar las acciones que quería?**

Muy difícil

Difícil

Normal

Fácil

Muy fácil

**8. Establezca el nivel de dificultad de la aplicación para usted**

Muy difícil

Difícil

Normal

Fácil

Muy fácil

**9. Puntúe en una escala del 1 al 5 cómo de intuitiva ha sido la aplicación para usted. (1=“Muy poco intuitiva”, 5=“Muy intuitiva”**

1

2

3

4

5

**10. En términos generales, ¿está usted satisfecho con la aplicación?**

No, nada

Poco satisfecho

Satisfecho

Bastante satisfecho

Muy satisfecho

**Figura 5.1** Cuestionario desarrollado para la evaluación objetiva de la aplicación

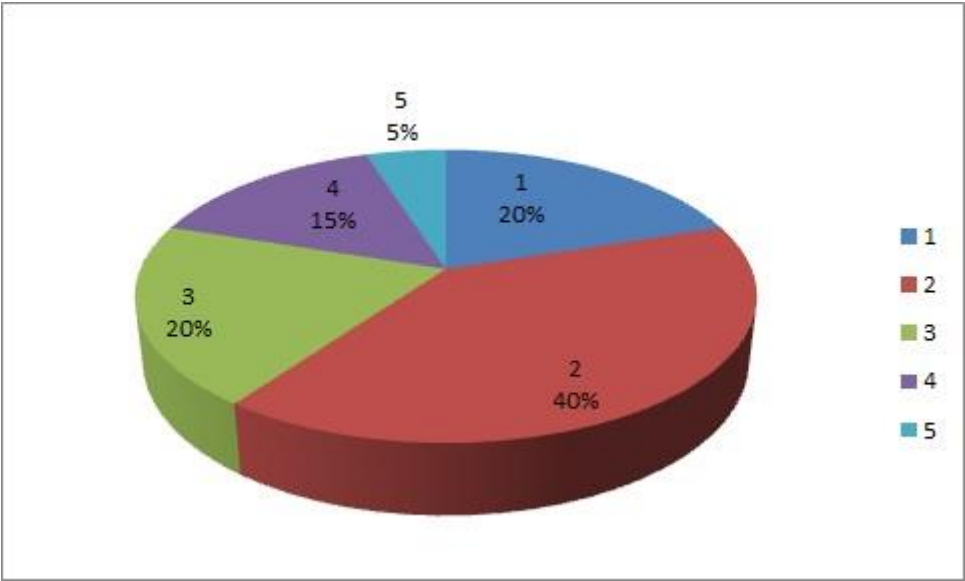
A partir de las respuestas de los diferentes usuarios a los que se les ha entregado el cuestionario anterior, se han obtenido las estadísticas para el análisis de los resultados de la evaluación que se muestran en la siguiente sección de la memoria.

## 5.2 Resultados de la evaluación

Como se ha comentado anteriormente, el test de evaluación de la aplicación ha sido realizado por un total de 20 usuarios potenciales elegidos al azar a los que se les permitió navegar por la aplicación libremente para que descubrieran por ellos mismos las funcionalidades de la aplicación por un tiempo aproximado de 20 minutos.

En las sucesivas figuras se muestra el resultado de la evaluación. Para un estudio más exacto, se han calculado el valor medio, la mediana, el valor máximo y mínimo así como la varianza para cada una de las preguntas de las que se compone el cuestionario.

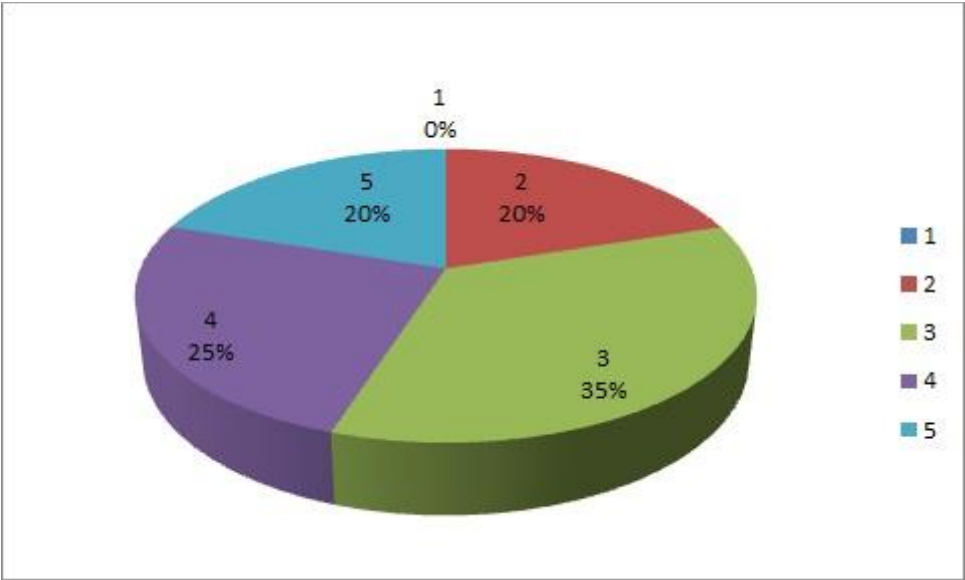
**Pregunta 1.- Puntúe en una escala del 1 al 5 su experiencia previa usando dispositivos móviles.**



**Figura 5.2 Respuesta de los participantes a la pregunta 1**

Valor medio	Mediana	Valor máximo	Valor mínimo	Varianza
2,45	2	5	1	1,25

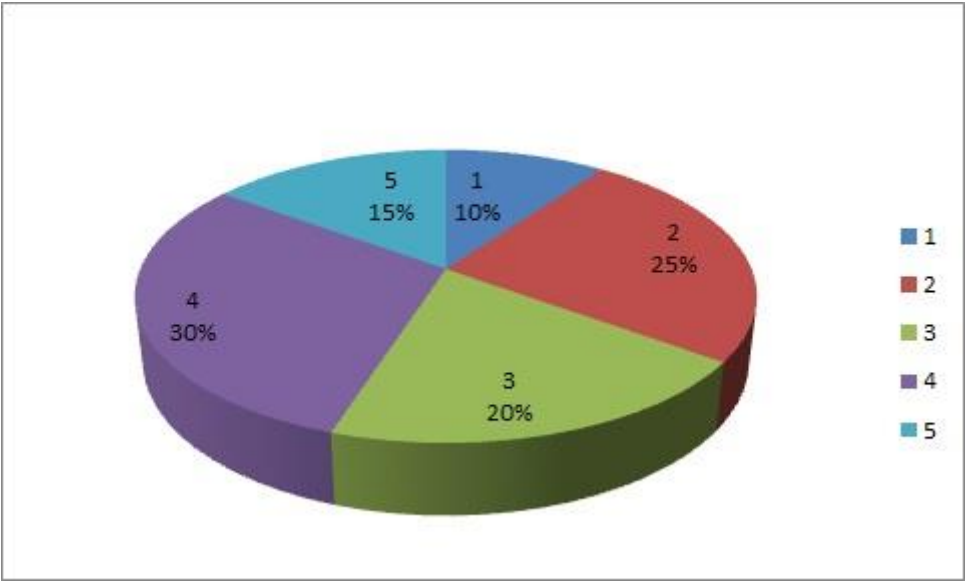
**Pregunta 2.- ¿Con qué frecuencia utiliza su smartphone al cabo de un día?**



**Figura 5.3 Respuesta de los participantes a la pregunta 2**

Valor medio	Mediana	Valor máximo	Valor mínimo	Varianza
3,45	3	5	2	1,05

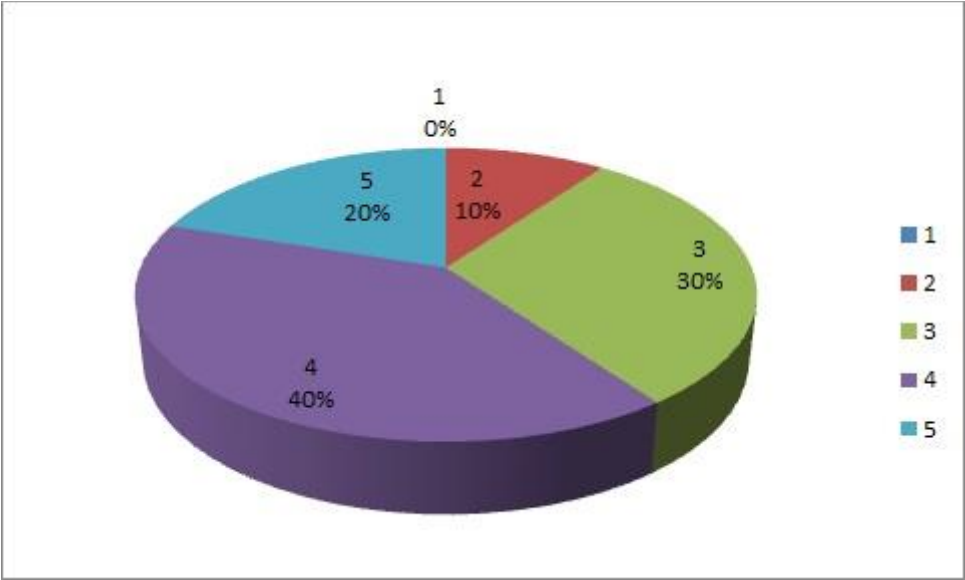
**Pregunta 3.- Puntúe en una escala del 1 al 5 su experiencia previa usando interfaces de voz o sistemas similares**



**Figura 5.4 Respuesta de los participantes a la pregunta 3**

Valor medio	Mediana	Valor máximo	Valor mínimo	Varianza
3,15	3	5	1	1,53

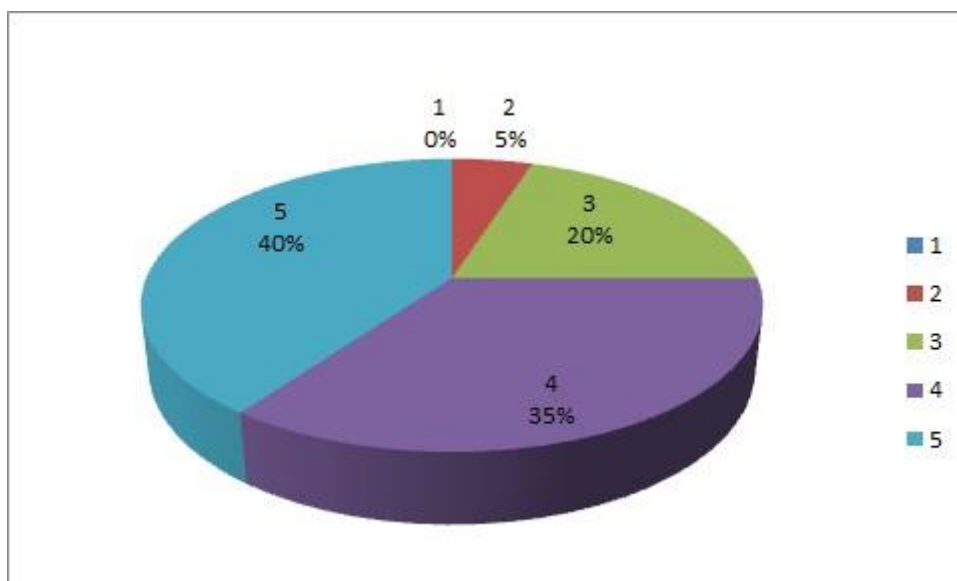
**Pregunta 4.- ¿Qué tal le ha entendido la aplicación?**



**Figura 5.5 Respuesta de los participantes a la pregunta 4**

Valor medio	Mediana	Valor máximo	Valor mínimo	Varianza
3,7	4	5	2	0,81

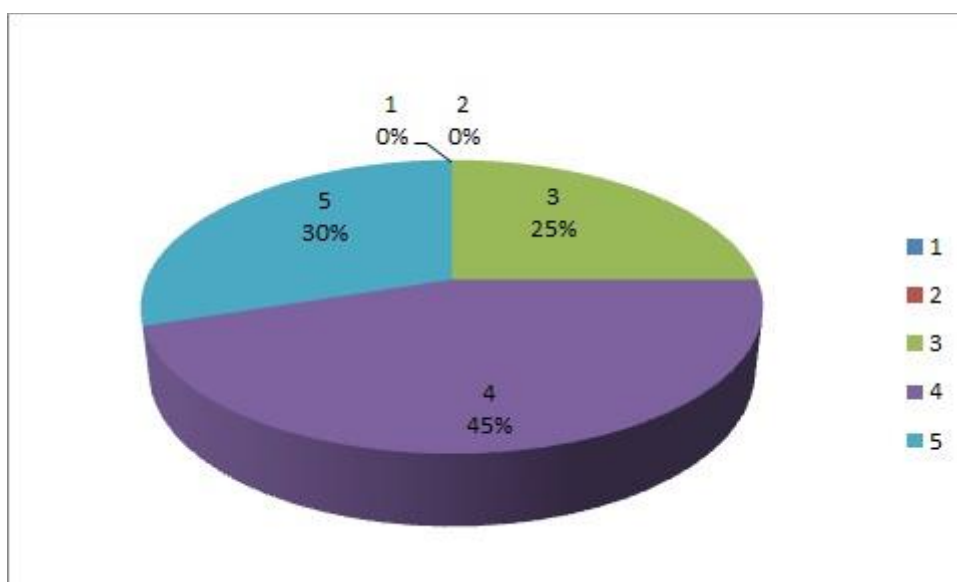
**Pregunta 5.- ¿Qué tal ha entendido los mensajes generados por la aplicación?**



**Figura 5.6 Respuesta de los participantes a la pregunta 5**

Valor medio	Mediana	Valor máximo	Valor mínimo	Varianza
4,1	4	5	2	0,79

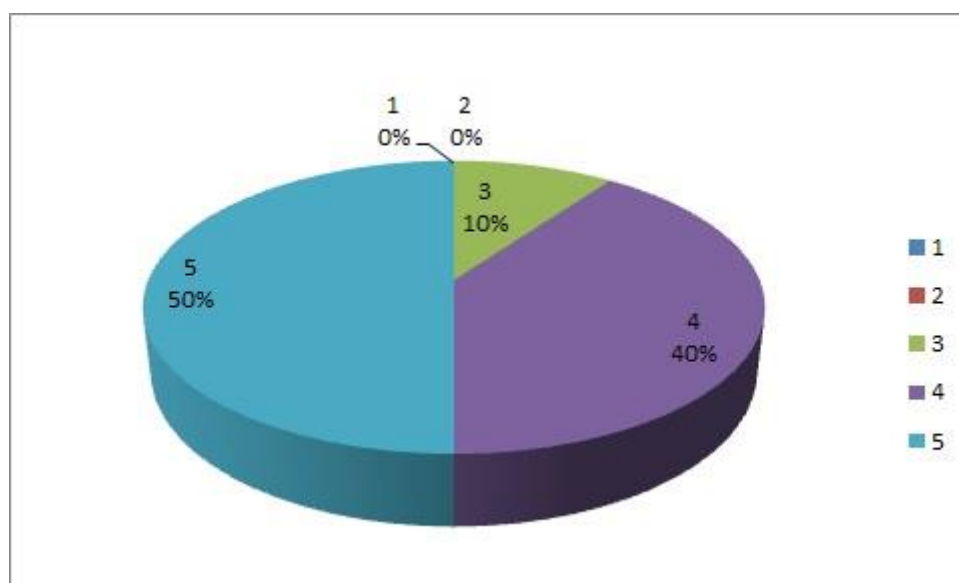
**Pregunta 6.- En su opinión, la navegación por la aplicación fue...**



**Figura 5.7 Respuesta de los participantes a la pregunta 6**

Valor medio	Mediana	Valor máximo	Valor mínimo	Varianza
4,05	4	5	3	0,55

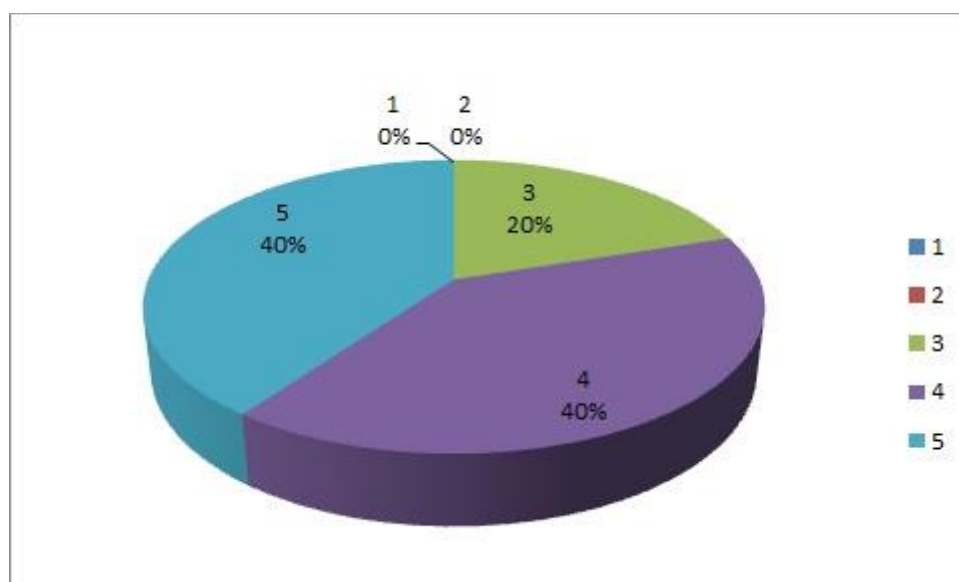
**Pregunta 7.- ¿Le ha sido fácil realizar las operaciones que quería?**



**Figura 5.8 Respuesta de los participantes a la pregunta 7**

Valor medio	Mediana	Valor máximo	Valor mínimo	Varianza
4,4	5	5	3	0,44

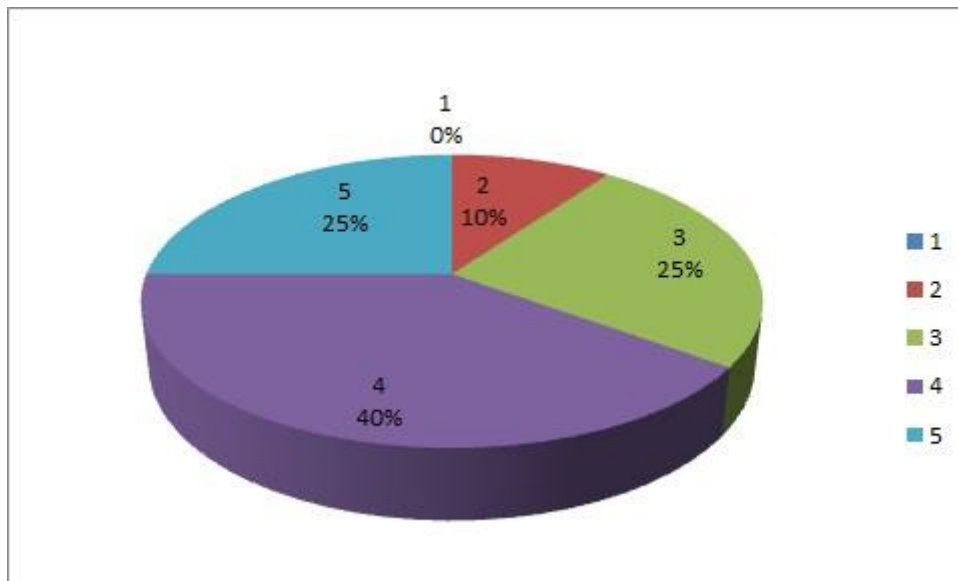
**Pregunta 8.- Establezca el nivel de dificultad de la aplicación para usted.**



**Figura 5.9 Respuesta de los participantes a la pregunta 8**

Valor medio	Mediana	Valor máximo	Valor mínimo	Varianza
4,2	4	5	3	0,56

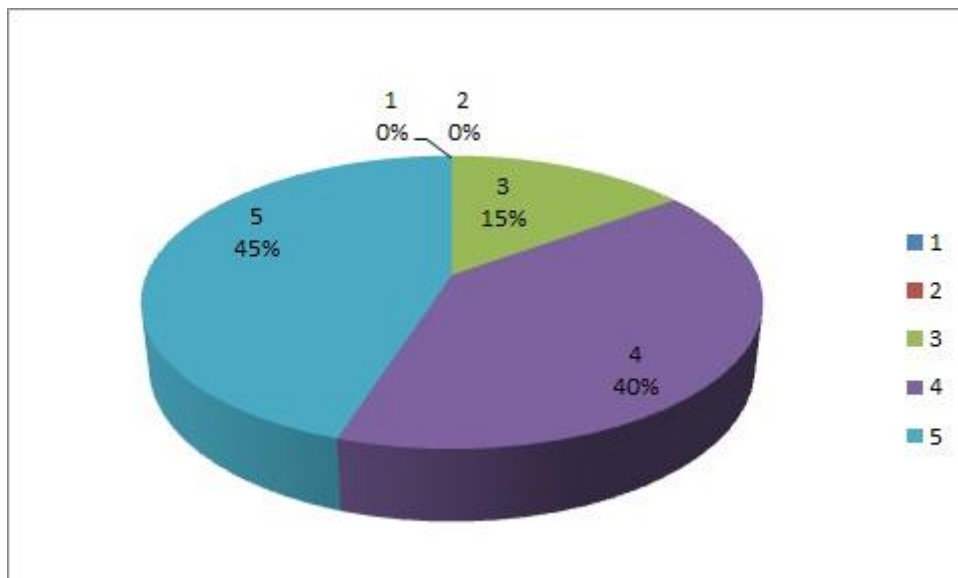
**Pregunta 9.-** Puntúe en una escala del 1 al 5 cómo de intuitiva ha sido la aplicación para usted.



**Figura 5.10** Respuesta de los participantes a la pregunta 9

Valor medio	Mediana	Valor máximo	Valor mínimo	Varianza
3,8	4	5	2	0,86

**Pregunta 10.-** En términos generales, ¿está usted satisfecho con la aplicación?



**Figura 5.11** Respuesta de los participantes a la pregunta 10

Valor medio	Mediana	Valor máximo	Valor mínimo	Varianza
4,3	4	5	3	0,51

Del análisis de las respuestas proporcionadas por los participantes en la evaluación de la aplicación se pueden extraer las siguientes conclusiones:

- A pesar de que la mayoría de los participantes no tenía excesiva experiencia previa en lo que se refiere al uso de dispositivos móviles inteligentes, la aplicación es mayoritariamente valorada como sencilla de utilizar y de navegar por ella debido a que los usuarios la consideran como muy intuitiva.
- Además, también se observa que la comunicación entre el usuario y la aplicación es muy buena, ya que tanto los mensajes de voz que el usuario produce como los mensajes generados por la propia aplicación son entendidos sin grandes dificultades.
- Por último, resaltar que el grado de satisfacción con la aplicación de la mayoría de los participantes es muy elevado, llegando incluso a obtener en esta pregunta la mejor valoración de la aplicación.



# Capítulo 6

## Conclusiones y trabajo futuro

En este capítulo se hace un balance del trabajo realizado en el presente Trabajo Fin de Grado. Partiendo de la revisión de los resultados logrados se comprueba que se han cumplido los objetivos fijados inicialmente y se exponen las conclusiones obtenidas. Para finalizar, se presentan los posibles trabajos futuros que se podrían efectuar sobre la aplicación desarrollada con el fin de mejorar su eficiencia así como el número de funcionalidades que cubriría.

### 6.1 Conclusiones

En este Trabajo Fin de Grado se ha desarrollado una aplicación en Android basada en el reconocimiento de voz. Se trata de una aplicación que permite a los usuarios manejar su agenda telefónica, realizar llamadas y enviar mensajes mediante mandatos de voz. Asimismo, también es posible realizar todas las funciones citadas anteriormente mediante el teclado del dispositivo móvil del usuario.

Los servicios ofrecidos por la aplicación desarrollada se reparten en módulos por los cuales se conduce al usuario en función de las decisiones que haya tomado en cada intervención. Estos módulos están interconectados de tal forma que el camino más largo para acceder de un módulo a otro es pasar únicamente por otro intermedio. Este hecho agiliza sobre manera la aplicación.

Desde el **módulo Contactos** el usuario puede acceder al resto de módulos de la aplicación de manera directa. Se trata del núcleo central sobre el que gira la aplicación, ya que es el primero al que accede el usuario después de registrarse o validarse, y el último antes de salir de la aplicación.

Además de dar acceso al resto de módulos de la aplicación, el módulo **Contactos** también tiene una funcionalidad propia. Desde él se accede a la agenda telefónica del usuario, mostrándose en una lista ordenada alfabéticamente.

Otro servicio que ofrece el sistema desarrollado con su **módulo Frecuentes** es la ordenación de la lista, mencionada anteriormente, según el número de conversaciones que se ha mantenido con cada uno de los contactos. De esta manera, el usuario puede visualizar en primer lugar aquél contacto con el que más veces ha interactuado. Para generar esta lista, se han considerado únicamente las comunicaciones establecidas por parte del usuario; es decir, si el contacto en cuestión se comunica con el usuario el orden de la lista no se ve alterado.

Además, mediante los módulos **Añadir**, **Editar** y **Borrar Contacto**, el usuario tiene la posibilidad de realizar todos los cambios que considere oportunos en su agenda telefónica. Estos cambios los puede realizar tanto mediante mandatos de voz como mediante el teclado virtual que incorpora el dispositivo móvil. En el caso del módulo **Borrar Contacto**, antes de realizar la acción de borrado propiamente dicha se reclama una última verificación por parte del usuario. De esta manera se evita que se eliminen de la agenda contactos por error.

En el **módulo Bandeja Mensajes**, el usuario dispone de todos los mensajes de texto enviados y recibidos de manera ordenada según la fecha de envío o recibo. Además, sirve de acceso directo al **módulo Enviar Mensaje** desde donde se realiza propiamente la acción de envío. En caso de que se reciba un mensaje del contacto con el que estamos manteniendo la conversación, la aplicación, si se tiene habilitada la opción de audio, lee el contenido del mismo sin necesidad de interacción por parte del usuario. En caso de que llegue un mensaje de texto de otro contacto, el **módulo Recibir Mensaje** genera una notificación con el logo de la aplicación indicando el nombre del contacto en cuestión. Al desplegar el menú y pulsar sobre la notificación, se accede a la ventana de mensajes del contacto.

Para que los usuarios puedan personalizar algunos aspectos de la aplicación cuando lo estimen oportuno, se ha desarrollado el **módulo Configuración**. En él, se puede decidir si se quiere que la aplicación interactúe de forma oral o no con el usuario. Estas comunicaciones se llevan a cabo tanto si los mandatos se realizan mediante la voz como si se realizan mediante el teclado virtual del dispositivo móvil. Así mismo, también se le da la opción al usuario de la aplicación desarrollada a cambiar la contraseña con la que se registró en la aplicación. Para llevar a cabo esta acción es necesario que se introduzca la antigua contraseña así como la nueva de forma repetida. Con motivo de otorgarle una mayor seguridad a la contraseña, ésta debe cumplir unos requisitos mínimos de seguridad, como son una longitud mínima de 6 caracteres y que contenga mayúsculas, minúsculas, números, signos de puntuación, etc.

Por último, el usuario de la aplicación desarrollada también tiene la opción de realizar llamadas telefónicas a cualquier contacto existente en su agenda telefónica. Esta acción se lleva a cabo en el **módulo Llamar**.

Para poder asignar a la aplicación desarrollada todas las funcionalidades relativas a los sistemas de diálogo que incorpora el sistema operativo Android, ha sido necesario realizar un estudio completo de todas las versiones existentes, examinando tanto las funcionalidades que ofrecían como su cuota de mercado.

La infraestructura y los componentes de reconocimiento y síntesis de voz necesarios para crear y utilizar la presente aplicación han sido los proporcionados por Google. Las conclusiones a destacar de estos servicios son la buena identificación de la información entrante por parte del reconocedor de voz, así como la buena asimilación de los mensajes de voz producidos por la aplicación.

Además, tal y como se ha demostrado en el estado del arte del presente documento, se trata del sistema operativo puntero para los dispositivos móviles, lo que incrementa exponencialmente las posibilidades de difusión del sistema implementado.

Del proceso de implementación del presente sistema, como sugerencias de mejora para el desarrollo de aplicaciones con interfaces orales, cabe destacar las siguientes consideraciones de diseño:

- Debido a que los usuarios potenciales de la aplicación son personas inexpertas con las nuevas tecnologías, es recomendable ofrecer numerosos mecanismos de ayuda y sistemas de realimentación, que indiquen al usuario qué ocurre en cada momento, además de tener en cuenta al diseñar la aplicación que las locuciones no sean excesivamente largas, los menús no tengan un gran número de opciones y los niveles en la estructura de navegación sean los mínimos posibles.
- Para que el usuario se habitúe al modo de funcionamiento del servicio, es importante mantener la coherencia y homogeneidad a lo largo de los diálogos, manteniendo un procedimiento de actuación similar en distintas secciones.
- Otra importante característica de la aplicación es que admite diversos mandatos para realizar la misma acción, lo que le otorga un grado de flexibilidad y dinamismo bastante notable.

Como conclusión final, los **objetivos** planteados en un principio **se han cumplido**. Con la realización de este Trabajo Fin de Grado se ha llevado a cabo un estudio completo de los sistemas de diálogo y de los sistemas operativos móviles, haciendo especial hincapié en Android, para poder así aplicar todas las posibilidades de las interfaces y ventajas que

ofrece en el desarrollo de la nueva aplicación, e implementar las funcionalidades necesarias con el fin de lograr una comunicación lo más sencilla y efectiva posible.

Mediante la aplicación desarrollada se ha elaborado un sistema de acceso y gestión a la agenda de contactos, así como de envío de mensajes de texto por parte de los usuarios. Estas acciones se pueden realizar tanto de forma oral como utilizando el teclado virtual del que dispone el dispositivo móvil. De este modo, se proporciona flexibilidad y facilidad de acceso, ya que en caso de que no se pueda hacer uso del acceso oral debido a que no se disponga de Internet, el usuario puede realizar las mismas acciones mediante el teclado. Además, estas acciones no requieren de gran experiencia, ya que la aplicación es muy intuitiva.

Finalmente, otro de los objetivos más importantes cumplidos es el de facilitar la accesibilidad a los servicios proporcionados por esta aplicación a usuarios inexpertos de las nuevas tecnologías, así como a personas con discapacidades visuales, gracias a la opción de aumentar o disminuir tanto como se quiera el tamaño de letra mediante un simple mandato de voz. Esta aplicación puede promover un mayor conocimiento, acercamiento e integración de estos colectivos dentro del universo de las nuevas tecnologías

En la actualidad, la gran mayoría de las aplicaciones orales existentes permiten únicamente realizar una serie de acciones por medio de la voz, sin existir la reciprocidad de que la aplicación se comunique también de forma oral, para realizar acciones tan sencillas como leerles un mensaje o advertirles de alguna incidencia.

La complejidad de la aplicación es elevada debido a la diversidad de cada una de las funciones implementadas, ya que en cada uno de los módulos y submódulos el diseño de las rutinas varía. Además, se debe tener en cuenta la dificultad que supone el hecho de incorporar un mecanismo de seguridad adicional como es el cifrado de la base de datos.

En definitiva, gracias a esta aplicación con acceso oral, los usuarios disponen de un sistema útil y accesible por todos que les proporciona de forma fácil y eficaz el acceso a las funcionalidades básicas de un dispositivo móvil.

## 6.2 Trabajo futuro

Tras la finalización de este Trabajo Fin de Grado, se detallan a continuación los puntos adicionales sobre los que se podría trabajar con el fin de obtener mayores prestaciones en la aplicación desarrollada. Estos trabajos futuros se pueden dividir en dos grandes bloques.

El primer bloque consiste en la ampliación de las funcionalidades ya existentes con el fin de mejorar el rendimiento de la aplicación:

### **Comunicación con la aplicación**

- Incorporar un reconocedor de voz que sea capaz de detectar el estado de ánimo del usuario de la aplicación y en función de éste actuar de una manera, calmándolo en aquellos momentos en los que advierta que el usuario está tenso, y comunicándose de forma más distendida en aquellos momentos que detecte que el usuario está de mejor humor.
- Conseguir que el servicio de reconocimiento de voz trabaje también cuando la aplicación no tiene acceso a Internet. De esta manera, el servicio estaría disponible en todo momento, independientemente de si el usuario dispone de tráfico de datos o no.

### **Comunicación con otros usuarios**

- Permitir la comunicación entre varios contactos de la agenda, pudiéndose realizar comunicaciones entre más de dos usuarios. Así mismo, también se podría incluir la posibilidad de enviar ficheros de audio e imágenes.

### **Seguridad**

- En lo que se refiere a aspectos de seguridad, se podría incluir un mecanismo de reconocimiento de retina o de huella dactilar. De esta manera, se evitaría el acceso a la aplicación incluso cuando el atacante disponga de la contraseña.

El segundo bloque de trabajos futuros consiste en la adición de funcionalidades globales del sistema, con los que se cubrirían aspectos que con la versión actual no se realizan:

### **Idiomas**

- Incluir la posibilidad de utilizar la aplicación para varios idiomas además de para el español, como actualmente está definido. Este hecho ampliaría considerablemente las fronteras de difusión de la aplicación.
- Permitir la comunicación con otros usuarios en distintas lenguas. Se podría emplear un traductor para que el usuario se comunicase con otros usuarios en otras lenguas. Con esta funcionalidad, se eliminarían las fronteras existentes para aquellas personas que no tienen conocimientos de otros idiomas además del materno.

### **Comunicación con otros usuarios vía Internet.**

- Actualmente la aplicación está diseñada para enviar únicamente mensajes de texto. Con la creación de un servidor, se podría realizar una comunicación vía Internet entre usuarios de la aplicación. Con esta nueva funcionalidad, se ahorraría el coste del envío de mensajes. Además, se podría incorporar métodos de cifrado y firma de mensajes para otorgarle una mayor robustez a esta funcionalidad.

# Presupuesto

En este apartado se presenta el coste total de realizar el Trabajo Fin de Grado. Para determinar dicha cifra se contempla la duración de las distintas fases y tareas y se incluye un desglose de costes de personal, costes de material y costes totales.

## 1. Tareas

### Fase 1: Planificación.

- Estudio de los sistemas operativos móviles.  
🕒 Duración: 7 días.
- Estudio del sistema operativo Android.  
🕒 Duración: 7 días.
- Estudio de los sistemas de diálogo.  
🕒 Duración: 7 días.
- Planificación y análisis de requisitos de la aplicación.  
🕒 Duración: 7 días.
- Estudio de las tecnologías necesarias.  
🕒 Duración: 30 días.

### Fase 2: Desarrollo.

- Análisis y diseño inicial.  
🕒 Duración: 7 días.
- Implementación del sistema.  
🕒 Duración: 45 días.
- Pruebas unitarias.  
🕒 Duración: 15 días.
- Pruebas de integración.  
🕒 Duración: 7 días.

- Evaluación de la aplicación.

🕒 Duración: 7 días.

### **Fase 3: Documentación**

- Memoria del Trabajo Fin de Grado.

🕒 Duración: 15 días.

- Preparación de la presentación.

🕒 Duración: 7 días.

## **2. Recursos**

- Recursos de Hardware.

- Ordenador de mesa: 900€
- Monitor: 150€
- Smartphone con Android 2.3 o superior: 300€
- USB de 8GB: 15€

- Recursos de Software.

- Plataforma Eclipse con plugins de Android: 0€
- Paquete Microsoft Office 2010: 230€
- Adobe Photoshop CS6: 285€
- Adobe Reader 9: 0€
- Editor de programación Notepad ++: 0€
- Dropbox: 0€

- Recursos humanos.

En la realización de este Trabajo Fin de Grado ha participado dos personas, el tutor del proyecto y el desarrollador.

Las jornadas dedicadas al proyecto han sido de una media de 5 horas, considerando semanas de 7 días laborables.

- Coste del desarrollador: 22€/hora.



### 3. Resumen de costes

El coste de recursos humanos de la aplicación se resume en la Figura 7.1.

TAREA	DÍAS	IMPORTE
FASE1	58	7.656€
FASE2	81	10.692€
FASE3	22	2.904€
Subtotal	161	21.252€

**Figura 7.1 Detalle de costes de recursos humanos del Trabajo Fin de Grado**

El coste total de la aplicación se presenta en la Figura 7.2.

CONCEPTO	IMPORTE
Recursos Humanos	21.252€
Recursos Software	515€
Recursos Hardware	1.365€
Subtotal	23.132€
(21% IVA)	4.857€
<b>TOTAL</b>	<b>27.989€</b>

**Figura 7.2 Detalle del coste total del Trabajo Fin de Grado**

El presupuesto total de este Trabajo Fin de Grado asciende a la cantidad de **VEINTISIETE MIL NOVECIENTOS OCHENTA Y NUEVE EUROS**.

Madrid, a 24 de Septiembre de 2013

Fdo. Carlos Molina López





# Glosario

- **Android:** es un sistema operativo basado en Linux, diseñado principalmente para dispositivos móviles con pantalla táctil como teléfonos inteligentes o tabletas, inicialmente desarrollado por Android Inc., que Google respaldó económicamente y más tarde compró en 2005.
- **API** (Application Programming Interface): es un conjunto de funciones y procedimientos que provee un sistema operativo, una aplicación o una biblioteca, que permiten leer o añadir contenido de un software en sus sitios web.
- **ASR** (*Automatic Speech Recognition*): es la capacidad de una máquina o un programa para identificar palabras y frases en el lenguaje hablado y convertirlos a un formato legible por la máquina.
- **CBC** (*Cipher Block Chaining*): es un modo de operación para un cifrado de bloques (se codifica la secuencia de bits como una sola unidad o bloque con un cifrado de clave que se aplica a todo el bloque). Este modo emplea lo que se conoce como vector de inicialización (IV) de una cierta longitud. Una de sus principales características es que el descifrado de un bloque de texto codificado depende de todos los bloques de texto cifrados con anterioridad.
- **Dalvik:** es la máquina virtual que utiliza la plataforma para dispositivos móviles Android. Está optimizada para requerir poca memoria y está diseñada para permitir ejecutar varias instancias de la máquina virtual simultáneamente, delegando en el sistema operativo subyacente el soporte de aislamiento de procesos, gestión de memoria e hilos.
- **HTML5:** es la quinta revisión del lenguaje de marcado HTML. Esta revisión especifica dos variantes de sintaxis para HTML: la clásica, conocida como HTML5 y otra variante, XHTML, conocida como sintaxis XHTML5. A pesar de que se encuentra en modo experimental, ya es usado por desarrolladores web.
- **Java:** es un lenguaje de programación desarrollado en el año 1995 por James Gosling de Sun Microsystems específicamente para entornos de red como Internet. Se trata de un lenguaje orientado a objetos, lo que significa que sus programas se construyen con módulos de código. A diferencia de otros lenguajes de programación, Java es un lenguaje interpretado, y por lo tanto, tiene más tiempo de ejecución.

- **Linux:** es un sistema operativo de software libre, lo que significa que todo su código fuente puede ser utilizado, modificado y redistribuidos libremente bajo los términos de la GPL (Licencia Pública General de GNU), y otra serie de licencias.

- **OpenGL:** es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzca gráficos 2D y 3D. La interfaz consiste en más de 250 funciones diferentes que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como puntos, líneas y triángulos.

- **PKCS5:** es un sistema de relleno (*padding*) para cifrados por bloque que consiste en rellenar tantos bytes como se necesiten. El valor de los bytes será el mismo que el número de los bytes introducidos. Este sistema únicamente es válido para cifrados con bloques de tamaño menor que 256, que es el valor máximo que puede tener un byte.

- **Plugin:** es un componente de software que añade una característica específica a una aplicación ya existente. Esta aplicación adicional es ejecutada por la principal, interactuando por medio de la API. Los ejemplos más comunes son los plugins utilizados en navegadores web para añadir nuevas características, tales como motores de búsqueda o antivirus.

- **PostgreSQL:** es un sistema libre de gestión de bases de datos relacional orientado a objetos. Es desarrollado por un grupo de voluntarios apoyados por organizaciones comerciales. Este sistema permite el acceso de varios procesos a una misma tabla sin necesidad de bloqueos.

- **SDK** (*Software Development Kit*): es un kit de desarrollo de software. En el caso del SDK de Android, es el proceso por el cual las nuevas aplicaciones se crean para este sistema operativo móvil. Incluye un conjunto de herramientas de desarrollo como bibliotecas, un depurador, un terminal emulador basado en QEMU, código de ejemplo, documentación, etc.

- **SQLite:** es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una pequeña biblioteca escrita en C. Es un proyecto de dominio público creado por Richard Hipp. A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor SQLite no es un proceso independiente con el que el programa principal se comunica.

- **TTS** (*Text To Speech*): es la generación de redes inalámbricas por medios automáticos de una voz artificial que genera el sonido producido por una persona al leer un texto cualquier en voz alta o una voz artificial. Es decir, son sistemas que permiten la conversión de textos en voz sintética.

- **USB** (*Universal Serial Bus*): es un estándar industrial desarrollado en el año 1990 que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre ordenadores y periféricos y dispositivos electrónicos.
- **W3C** (*World Wide Web Consortium*): es una asociación internacional formada por organizaciones miembros del consorcio, personal y el público en general, que trabajan conjuntamente para desarrollar estándares web. W3C pretende guiar la web hacia su máximo potencial a través del desarrollo de protocolos y pautas que aseguren el crecimiento futuro de la web.
- **XML** (*eXtensible Markup Language*): es un lenguaje de marcas desarrollado por W3C, utilizado para almacenar datos en forma legible. Se trata de una adaptación y simplificación del lenguaje SGML y permite definir la gramática de lenguajes específicos para estructurar grandes documentos.

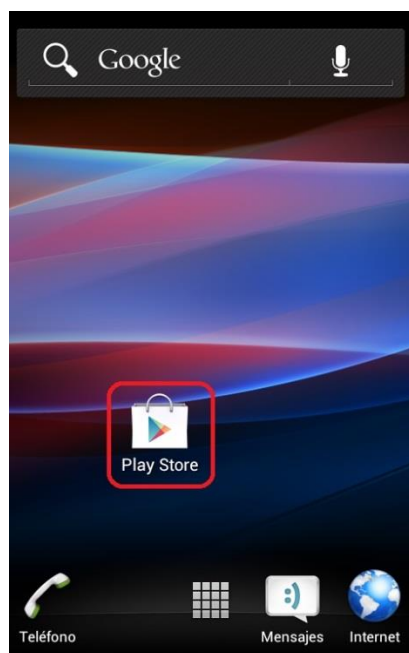


# Anexo

## Manual de instalación

Las aplicaciones móviles del sistema operativo Android se encuentran disponibles para la descarga en la tienda oficial, Google Play Store. Con la simple aceptación de los términos de la aplicación se automatiza la descarga y la instalación de la misma. En este anexo se expone el manual de instalación de la aplicación, en donde se explicarán los detalles concernientes a la descarga e instalación del programa desarrollado.

Como se ha explicado anteriormente, Google Play Store es el repositorio de aplicaciones de este sistema operativo. Todos los dispositivos de Android tienen acceso directo a la tienda oficial. En la Figura 9.1 se muestra una interfaz con el logo de Google Play Store marcado en rojo.



**Figura 9.1** Interfaz de un dispositivo Android con acceso a Google Play Store

Una vez se haya pulsado sobre dicho acceso directo, el usuario accede a la tienda oficial de Android. Como se puede observar en la Figura 9.2, desde esta página se puede acceder a distintos tipos de programas: aplicaciones, películas, libros, juegos, música, etc. Para descargar la aplicación desarrollada, el usuario deberá pulsar sobre la opción ‘Aplicaciones’.



**Figura 9.2 Programas ofrecidos por Google Play Store**

A continuación, para realizar una búsqueda más optimizada, es necesario acceder al menú de ‘Categorías’. Una vez en él, Figura 9.3, aparecerán las distintas categorías por las que se agrupan las aplicaciones alojadas en Google Play Store. En el caso de la aplicación desarrollada, su categoría es ‘Comunicación’.



**Figura 9.3 Búsqueda en Google Play Store por categoría**



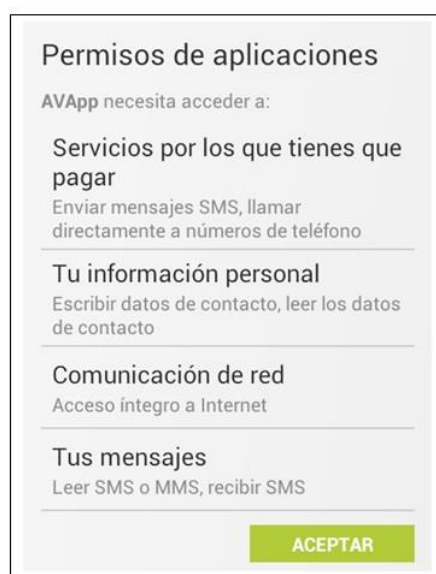
Para agilizar la búsqueda de la aplicación desarrollada, será necesario pulsar sobre el botón con forma de lupa que aparece en el margen superior derecho de la pantalla. A continuación se deberá escribir el nombre de la aplicación que se desea buscar. Para el programa desarrollado se ha elegido el nombre ‘AVApp’.

Una vez realizada la búsqueda, se desplegará una lista con todas las aplicaciones coincidentes. A continuación el usuario deberá pulsar sobre la aplicación que desea instalar, y accederá a la interfaz que se muestra en la Figura 9.4.



**Figura 9.4 Instalación de la aplicación desarrollada**

El último paso antes de que se lleve a cabo la descarga de la aplicación desarrollada será el de aceptar los permisos de la aplicación. Una vez pulsado el botón instalar, se desplegará una lista, Figura 9.5, con todos los permisos que se necesitan aceptar para su correcta instalación.



**Figura 9.5 Permisos de la aplicación desarrollada**

A continuación se iniciará el proceso de descarga de la aplicación. Como se muestra en la Figura 9.6, una vez se haya finalizado dicho proceso, se creará un acceso directo de la aplicación desarrollada. Pulsando sobre el icono, el usuario podrá disfrutar de todas las funcionalidades que ofrece.



**Figura 9.6 Acceso directo a la aplicación desarrollada**

# Bibliografía

## [ANDROIDCONTACT]

Disponible:

<http://android-contact-id-vs-raw-contact-id.blogspot.com.es/> [16 de julio de 2013]

## [ANDROIDEITY]

Disponible:

<http://androideity.com/2011/07/04/arquitectura-de-android/> [04 de julio de 2013]

## [ANDROIDLIBRE]

Disponible:

<http://www.elandroidelibre.com/2011/10/cinco-aplicaciones-de-control-por-voz-al-estilo-siri-de-apple-en-android.html> [30 de agosto de 2013]

## [ANDROIDZONE]

Disponible:

<http://androidzone.org/2013/05/historia-de-android-la-evolucion-a-lo-largo-de-sus-versiones/> [12 de agosto de 2013]

## [ASRANDROID]

Disponible:

<http://viralpatel.net/blogs/android-speech-to-text-api/> [28 de junio de 2013]

## [APK]

Disponible:

<http://androideity.com/2011/08/25/%C2%BFcomo-firmar-aplicaciones-android/>  
[20 de septiembre de 2013]

## [APPSTORE]

Disponible:

<http://store.apple.com/es/> [11 de agosto de 2013]

## [BOU10]

Boudriga, Noureddine. Security of mobile communications. 1ª edición. Boca Ratón: Auerbach Publications, 2010. ISBN: 978-0-8493-7941-3

## [BROADCAST]

Disponible:

[http://androidexample.com/Incomming\\_SMS\\_Broadcast\\_Receiver\\_-\\_Android\\_Example/index.php?view=article\\_discription&aid=62&aaid=87](http://androidexample.com/Incomming_SMS_Broadcast_Receiver_-_Android_Example/index.php?view=article_discription&aid=62&aaid=87)  
[21 de agosto de 2013]

## [BUR12]

Burnette, Ed. Programación Android. 2ª edición. Madrid: Anaya, 2012. ISBN: 978-84-415-2876-5

[CBC]

“*Structural Binary CBC Encryption Mode*”, Yi-Shiung Yeh, Ting-Yu Huang y Han-Yu Lin . National Chiao Tung University, Noviembre 2007. Disponible:  
[http://www.iis.sinica.edu.tw/page/jise/2009/200905\\_18.pdf](http://www.iis.sinica.edu.tw/page/jise/2009/200905_18.pdf)  
[09 de septiembre de 2013]

[DALVIK]

“*Dalvik Virtual Machine Architecture*”, David Ehringer. Marzo 2010. Disponible:  
[http://show.docjava.com/posterous/file/2012/12/10222640-The\\_Dalvik\\_Virtual\\_Machine.pdf](http://show.docjava.com/posterous/file/2012/12/10222640-The_Dalvik_Virtual_Machine.pdf) [09 de septiembre de 2013]

[CELULARIS]

Disponible:  
<http://www.celularis.com/software/historia-android/> [17 de agosto de 2013]

[DEVELOPERS]

Disponible:  
<http://developer.android.com/reference/packages.html> [16 de septiembre de 2013]

[ECLIPSE]

Disponible:  
<http://www.eclipse.org/downloads/> [03 de septiembre de 2013]

[GIR12]

Gironés, Jesús Tomás. El gran libro de Android. 2ª edición. Barcelona: Marcombo, 2012. ISBN: 978-84-267-1832-7

[GIR13]

“*El gran libro de Android*” Jesús Tomás Gironés. Universitat Politècnica de València, Enero 2013. Disponible:  
<http://www.androidcurso.com/index.php/recursos-didacticos> [06 de julio de 2013]

[GOOGLEPLAY]

Disponible:  
[https://support.google.com/googleplay/android-developer/answer/113468?hl=es&ref\\_topic=2365624](https://support.google.com/googleplay/android-developer/answer/113468?hl=es&ref_topic=2365624) [20 de septiembre de 2013]

[GRI07]

Griol, David. Desarrollo y evaluación de Diferentes Metodologías para la Gestión Automática del Diálogo. Tesis Doctoral. Universidad Politécnica de Valencia, 2007.

[HUR+05]

Lluís Hurtado, Fernando Blat, S. Grau, David Griol, Emilio Sanchís, Encarna Segarra, F. Torres. Sistema de diálogo para el Proyecto DIHANA. Procesamiento del Lenguaje Natural, ISSN 1135-5948, N°. 35, 2005, págs. 453-454.

[ITUNES]

Disponible:  
<http://www.apple.com/es/itunes/download/> [10 de agosto de 2013]

**[LAYOUT]**

Disponible:

[http://elbauldelprogramador.com/opensource/programacion-android-interfaz-grafica\\_23/](http://elbauldelprogramador.com/opensource/programacion-android-interfaz-grafica_23/) [04 de septiembre de 2013]

**[LLI06]**

Llisterri, J. (2006) "Introducción a los sistemas de diálogo", in LLISTERRI, J.-MACHUCA, M. J. (Eds.) *Los sistemas de diálogo*. Bellaterra - Soria: Universitat Autònoma de Barcelona, Servei de Publicacions - Fundación Duques de Soria (Manuals de la Universitat Autònoma de Barcelona, Lingüística, 45), págs. 11-21.

**[LOP05]**

López-Cózar, R., and Araki, M.: "Spoken, Multilingual and Multimodal Dialogue Systems: Development and Assessment" (Wiley, 2005).

**[MCT04]**

McTear, Michael F.: "Spoken Dialogue Technology" (Springer, 2004, 1 st Edition).

**[MKYONG]**

Disponible:

<http://www.mkylong.com/android/android-relativelayout-example/>  
[19 de agosto de 2013]

**[PIE12]**

Pieraccini, R., "The Voice in the Machine: Building Computers that Understand Speech" (The MIT Press, 2012).

**[PKCS5]**

"[PKCS#5: Password-based cryptography specification version 2.0](http://tools.ietf.org/html/rfc2898)", B. Kaliski, Network Working Group, Septiembre 2000. Disponible:  
<http://tools.ietf.org/html/rfc2898> [11 de septiembre de 2013]

**[PLUGIN]**

Disponible:

<https://developers.google.com/eclipse/docs/install-eclipse-4.3>  
[03 de septiembre de 2013]

**[PREZI]**

Disponible:

<http://prezi.com/vurapmevjfb-/sqlite/> [05 de septiembre de 2013]

**[OBS13]**

Disponible:

<http://www.obs-edu.com/noticias/2013/07/29/el-81-de-los-moviles-comprados-en-espana-en-2012-fueron-smartphones-2/> [27 de agosto de 2013]

**[OVISTORE]**

Disponible:

<http://store.ovi.com/> [28 de julio de 2013]

**[SDK]**

Disponible:

<http://developer.android.com/sdk/index.html> [03 de septiembre de 2013]

**[SGOLIVER]**

Disponible:

[http://www.sgoliver.net/blog/?page\\_id=3011](http://www.sgoliver.net/blog/?page_id=3011) [15 de septiembre de 2013]

**[SIRI]**

Disponible:

<http://www.apple.com/es/ios/siri/> [02 de septiembre de 2013]

**[SMSANDROID]**

Disponible:

<http://www.codeproject.com/Articles/463338/Sending-a-SMS-Message-from-an-Android-Application> [02 de agosto de 2013]

**[STACKOVERFLOW]**

Disponible:

<http://stackoverflow.com/> [13 de septiembre de 2013]

**[TAN98]**

Tanenbaum, Andrew S.; Woodhull, Albert S. Sistemas operativos: diseño e implementación. 2ª edición. Barcelona: Prentice Hall, 1998.

**[TTSANDROID]**

Disponible:

<http://www.androidhive.info/2012/01/android-text-to-speech-tutorial/>  
[05 de julio de 2013]

**[VIEW]**

Disponible:

<http://developer.android.com/reference/android/view/View.html> [06 de julio de 2013]

**[WINDOWSMARKET]**

Disponible:

<http://www.windowsmarketplace.com> [19 de julio de 2013]